Optimal Tree Methods

Ralph Rudd

A dissertation submitted to the Department of Actuarial Science, Faculty of Commerce, University of the Cape Town, in partial fulfilment of the requirements for the degree of Master of Philosophy.

May 22, 2014

Master of Philosophy specializing in Mathematical Finance,
University of the Cape Town,
Cape Town.

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the Degree of Master of Philosophy in the University of the Cape Town. It has not been submitted before for any degree or examination in any other University.

May 22, 2014

Abstract

Although traditional tree methods are the simplest numerical methods for option pricing, much work remains to be done regarding their optimal parameterization and construction. This work examines the parameterization of traditional tree methods as well as the techniques commonly used to accelerate their convergence. The performance of selected, accelerated binomial and trinomial trees is then compared to an advanced tree method, Figlewski and Gao's Adaptive Mesh Model, when pricing an American put and a Down-And-Out barrier option.

Acknowledgements

I thank my supervisor Professor Thomas McWalter for his insight and guidance, as well as my co-supervisor Professor David Taylor for directing a significant part of my education (most of which is still to come.)

I thank my parents, Ralph and Hanlie Rudd for personifying the idealized model of a parenting team. I thank Michelle McKerrell for her unending support during my pursuit of this degree and Hugo van Zijl for allowing me the use of his personal computer to generate the results.

Finally, I would like to thank HSBC Africa for the financial aid they provided to me during the 2013 academic year.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Overview

The subset of derivatives for which analytical pricing solutions exist is severely limited. For the majority of financial instruments, numerical pricing methods must be implemented.

On a high-level, the numerical methods for derivative pricing can be divided into two groups: lattice methods and Monte Carlo methods. Whereas Monte Carlo methods are based on the principle of randomly generating sample outcomes, lattice methods are concerned with discretizing both time and the state space of the underlying and then recursively solving the option price on the nodes of this generated lattice.

Traditionally, lattice methods are further divided into two groups: tree methods and finite difference methods. The most common tree structures are the binomial and trinomial models which, as their names imply, constrain the change in the underlying to two or three possible states respectively at each step. Finite difference methods are concerned with numerically solving the differential equation that models the derivative. Explicit finite difference methods are equivalent to trinomial tree models (Hull, 2010).

As pricing complicated financial instruments accurately and quickly allows for a competitive advantage in the financial industry, it is important to establish which techniques work best for which kind of products.

This work aims to addresses a subset of the question:

"Which is the optimal tree method?"

For the specific case of an American put option, Joshi (2007) makes significant inroads with regards to answering the above research question. In that work, the convergence of 220 different binomial trees is examined. These trees are constructed from a combination of 11 different parameterizations and 4 different acceleration techniques.

In Chan *et al.* (2009), this analysis is furthered by examining 128 combinations of trinomial trees.

1.2 Research Method and Aims

This work will extend the above analysis by re-implementing the best performing trees and comparing them to the Adaptive Mesh Model (AMM) developed by Figlewski and Gao (1999). The Adaptive Mesh Model is an evolution of the traditional trinomial tree structure created by overlaying a higher resolution tree onto an existing trinomial tree to obtain greater accuracy in the areas where the option value is distinctly non-linear.

Since the AMM is more complex to implement than the traditional tree methods, more flexibility is expected from the method. This will be tested by pricing a barrier option, when the barrier is close to the initial asset price, and comparing the convergence to that of the best performing binomial tree identified in Joshi (2007). Binomial trees are notoriously poor at valuing barrier options, and thus the simple modification proposed by Derman *et al.* (1995) will be made for a fairer comparison.

The work is divided into roughly four sections. The first deals with the derivation and construction of traditional tree models for the asset price. The second examines the potential sources of error and the techniques used to minimize them and accelerate the convergence of the models. The third examines the construction and parameterization of the adaptive mesh model and the final section examines the results of the implemented methods when used to price an American put option and a European Down-And-Out barrier option.

1.3 Notation

Throughout this work, the following notation is employed, with additional terms defined as needed:

$S(t)$	Value of a generic underlying asset at time t
$S_i(t)$	Value of a generic underlying asset at node i in the tree at time t
$V(t)$	Value of a generic derivative at time t
$V_i(t)$	Value of a generic derivative at node i in the tree at time t
N	The number of time-steps in the tree
t_j	The discrete time $j\Delta t$, as opposed to the continuous time point t
K	Strike price of the option
T	Maturity of the option
B	The level of the barrier
σ	Annualized volatility of the underlying asset
r	Annual continuously compounded risk-free rate of interest
p	Real-world probability of an upward movement
q	Risk-neutral probability of an upward movement
u	Multiplicative magnitude of an upward movement
d	Multiplicative magnitude of a downward movement
$n_i(t)$	Node i in the tree at time t

Note that the conventional numbering of tree nodes is followed here, where the nodes are numbered starting at the highest price for the underlying at that time to the lowest.

Chapter 2

Construction of Traditional Tree Methods

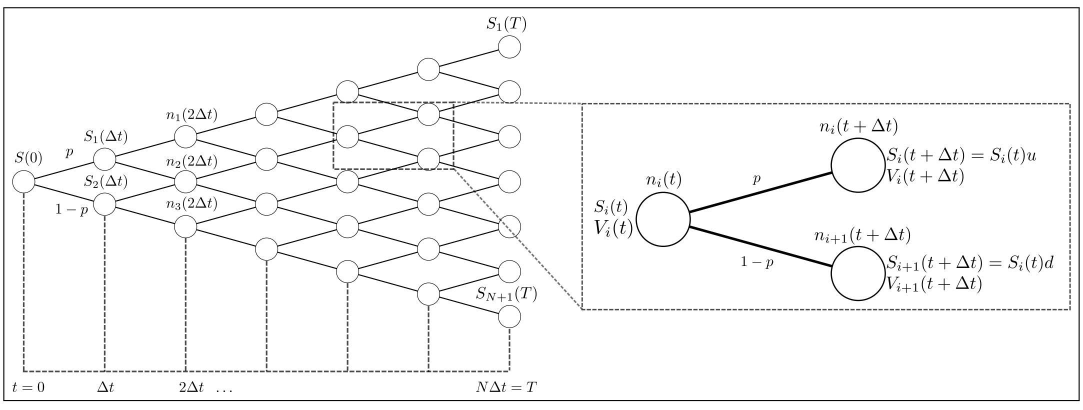
2.1 Binomial Trees

> *The so-called binomial model is really a family of models that, under surprisingly mild conditions, all converge in the limit to the Black-Scholes-Merton Model.*
>
> — Don M. Chance (Chance, 2007)

2.1.1 The Binomial Framework

Consider a market in discrete time and price space, consisting of a cash account M, an asset S and a derivative V written on the asset with European-style exercise,

Fig. 2.1: Illustration of multi-period binomial tree



maturity T and a payoff function $f(S(T))$. The market exists on the regular lattice $\pi = [0, \Delta t, 2\Delta t, \cdots, N\Delta t]$, created by dividing the time period $[0, T]$ into N discrete intervals of equal size, Δt.

Fundamentally, the binomial model is based on the assumption that at time $t_j = j\Delta t$, with $j = 0, \cdots, N-1$, the modelled asset price, $S(t_j)$, can move to only one of two values over the next time-step. At node i in a multi-period tree (see Figure 2.1), this can be written as

$$S(t_j + \Delta t) = \begin{cases} S_i(t_j + \Delta t) = S_i(t_j)u & \text{with probability } p, \\ S_{i+1}(t_j + \Delta t) = S_i(t_j)d & \text{with probability } 1-p, \end{cases} \tag{2.1}$$

where u is the multiplicative magnitude of an upward movement, d is the multiplicative magnitude of a downward movement and p is defined under the physical probability measure $\mathbb{P}$.

To price the derivative, V, written on $S(t_j)$ in this model, the following further assumptions are made regarding the market:

- The underlying distribution of $S(t_j)$ is stationary, i.e. u and d are time- and state-independent.
- Any fractional value of the asset can be bought or sold.
- The asset pays no dividends and makes no other distributions.
- There exists a known, short-term, continuously compounded risk-free rate, r, that is constant throughout time.
- Cash can be borrowed and invested at the same risk-free rate and in any amount.
- There are no bid-ask spreads, transaction costs or penalties for short-selling, i.e. the market is frictionless.
- There is no arbitrage in the market.

With these assumptions in place, it is possible to construct a dynamic replicating portfolio, $\vec{\psi}_i(t_j) = [\psi_i^S(t_j), \psi_i^M(t_j)]$, where $\psi_i^S(t_j)$ is the holding in the asset and $\psi_i^M(t_j)$ is the cash holding at tree-node i and discrete time t_j. Both holdings are transacted at t_j and held until $t_j + \Delta t$.

For the portfolio to replicate the derivative, $V(t_j)$, written on the asset, it must hold that

$$\psi_i^S(t_j)S(t_j)u + \psi_i^M(t_j)e^{r\Delta t} = V_i(t_j + \Delta t) \tag{2.2}$$

and

$$\psi_i^S(t_j)S(t_j)d + \psi_i^M(t_j)e^{r\Delta t} = V_{i+1}(t_j + \Delta t). \tag{2.3}$$

This is equivalent to stating that regardless of the movement of the asset over the interval the value of the portfolio must equal the value of the derivative at time $t_j + \Delta t$. Solving the above two equations simultaneously yields

$$\psi_i^S(t_j) = \frac{V_i(t_j + \Delta t) - V_{i+1}(t_j + \Delta t)}{uS_i(t_j) - dS_i(t_j)} \tag{2.4}$$

and

$$\psi_i^M(t_j) = e^{-r\Delta t}S_i(t_j)\frac{uV_{i+1}(t_j + \Delta t) - dV_i(t_j + \Delta t)}{uS_i(t_j) - dS_i(t_j)}. \tag{2.5}$$

Since the portfolio constituents are not altered during $(t_j, t_j + \Delta t)$, to ensure no-arbitrage the value of the portfolio at t_j must be the same as the value of the claim on the underlying. This yields

$$\begin{aligned} V_i(t_j) &= \psi_i^S(t_j)S_i(t_j) + \psi_i^M(t_j) \\ &= e^{-r\Delta t}\left[V_i(t_j + \Delta t)\left(\frac{e^{r\Delta t} - d}{u - d}\right) + V_{i+1}(t_j + \Delta t)\left(\frac{u - e^{r\Delta t}}{u - d}\right)\right] \\ &= e^{-r\Delta t}[qV_i(t_j + \Delta t) + (1 - q)V_{i+1}(t_j + \Delta t)], \end{aligned} \tag{2.6}$$

where

$$q = \frac{e^{r\Delta t} - d}{u - d}. \tag{2.7}$$

It would be useful to be able to consider q as a probability, but this requires the additional constraint that

$$0 \leq \frac{e^{r\Delta t} - d}{u - d} \leq 1. \tag{2.8}$$

To manipulate the above inequality, the sign of $u - d$ must be known. As u is the multiplicative magnitude of an upward movement, it is reasonable to constrain u to be greater than one and similarly to constrain d to be less than one. This allows the transformation of the above to

$$d \leq e^{r\Delta t} \leq u. \tag{2.9}$$

Although it is not immediately apparent, Equation (2.9) is the *no-arbitrage condition* for the binomial framework.

Theorem 2.1. *The multi-period binomial model is free of arbitrage if and only if*

$$d \leq e^{r\Delta t} \leq u. \tag{2.10}$$

Proof. This is an illustrative proof for the single-period model only and follows Taylor (2013). For the equivalent proof in the multi-period model (which requires a more exacting definition of arbitrage and self-financing portfolios), please see Björk (2004). The theorem is proven in two parts.

1. Let the model be free of arbitrage and let $d < u < e^{r\Delta t}$. Construct the portfolio $\vec{\psi}_1(t_0) = [-1, S_1(t_0)]$, which has value $V_P(t_0) = 0$ at $t_0 = 0$. At $t_1 = \Delta t$, the portfolio has one of two possible values

$$V_P(t_1) = \begin{cases} -S_1(t_0)u + S_1(t_0)e^{r\Delta t} & \text{with probability } p, \\ -S_1(t_0)d + S_1(t_0)e^{r\Delta t} & \text{with probability } 1-p, \end{cases} \tag{2.11}$$

both of which are always greater than 0 with strictly positive probability. This violates the no-arbitrage assumption. (The proof for $e^{r\Delta t} < d < u$ is similar.)

2. Let $d \leq e^{r\Delta t} \leq u$ hold. It is required to show that this ensures the absence of arbitrage. Construct an arbitrary portfolio with $V_P(t_0) = 0$, thus $\vec{\psi}_1(t_0) = [\psi_1(t_0), -\psi_1(t_0)S_1(t_0)]$. This portfolio can take on one of two possible values at $t_1 = \Delta t$,

$$V_P(t_1) = \begin{cases} \psi_1(t_0)S_1(t_0)u - \psi_1(t0)S_1(t_0)e^{r\Delta t} & \text{with probability } p, \\ \psi_1(t_0)S_1(t_0)d - \psi_1(t0)S_1(t_0)e^{r\Delta t} & \text{with probability } 1-p. \end{cases} \tag{2.12}$$

Therefore arbitrage can only occur if $u > d > e^{r\Delta t}$, or if $e^{r\Delta t} > u > d$ in the $\psi_1(t_0) < 0$ case, both of which violate the initial assumption.

□

As shown above, if Theorem 2.1 is satisfied, q is neatly constrained to the interval $[0, 1]$ and this allows for the construction of a discrete probability measure $\mathbb{Q}$, where q is the probability of an upward movement of the asset and $1 - q$ is the probability of a downward movement.

Definition 2.2. A measure, $\mathbb{Q}$, is said to be risk-neutral if, under $\mathbb{Q}$, the discounted asset price is a martingale.

Since

$$\begin{aligned}\mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t}S(t_j+\Delta t)|S_i(t_j)] &= e^{-r\Delta t}(quS_i(t_j)+(1-q)dS_i(t_j))\\ &= e^{-r\Delta t}\left(\frac{e^{r\Delta t}-d}{u-d}uS_i(t_j)+\frac{u-e^{r\Delta t}}{u-d}dS_i(t_j)\right)\\ &= S_i(t_j), \end{aligned} \tag{2.13}$$

$\mathbb{Q}$ is clearly a risk-neutral measure and the existence of $\mathbb{Q}$ allows Equation (2.7) to be interpreted as *risk-neutral pricing*, since

$$\begin{aligned}V_i(t_j) &= e^{-r\Delta t}[qV_i(t_j+\Delta t)+(1-q)V_{i+1}(t_j+\Delta t)],\\ &= \mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t}V(t_j+\Delta t)|V_i(t_j)]. \end{aligned} \tag{2.14}$$

This pricing equation has the important characteristic that both investor risk-preferences and real-world probabilities are absent. It is now possible to calculate the expectation and variance of the asset without reference to the physical probability measure.

Remark 2.1.1 – A Note on Returns.
For clarity, it is important to differentiate between "return" and "rate of return". Return is defined as future value over current value, or one plus the simple rate of return,

$$\text{return} = \frac{\text{future price}}{\text{current price}} = 1 + \text{rate of return}.$$

Rate of return is the change in value divided by the current value,

$$\text{rate of return} = \frac{\text{future price} - \text{current price}}{\text{current price}}.$$

Unfortunately, these are often used interchangeably in the literature and the confusion can be compounded by the relationship

$$\ln(\text{return}) = \ln(1 + \text{rate of return}) \approx \text{rate of return},$$

when the rate of return is sufficiently small.

Given information until time t_j, the expected return and expected log-return under the risk-neutral measure $\mathbb{Q}$ are respectively

$$\mathbb{E}^{\mathbb{Q}}\left[\frac{S(t_j+\Delta t)}{S_i(t)}\middle| S_i(t_j)\right] = qu + (1-q)d \tag{2.15}$$

and

$$\mathbb{E}^{\mathbb{Q}}\left[\ln\left(\frac{S(t_j+\Delta t)}{S_i(t)}\right)\middle| S_i(t_j)\right] = q\ln u + (1-q)\ln d. \tag{2.16}$$

The variance of returns and log-returns are respectively

$$\begin{aligned}\mathbb{V}\mathrm{ar}^{\mathbb{Q}}\left[\frac{S(t_j+\Delta t)}{S_i(t_j)}\middle| S_i(t_j)\right] &= qu^2 + (1-q)d^2 - (qu+(1-q)d)^2 \\ &= (u-d)^2 q(1-q) \end{aligned} \tag{2.17}$$

and

$$\begin{aligned}\mathbb{V}\mathrm{ar}^{\mathbb{Q}}\left[\ln\left(\frac{S(t_j+\Delta t)}{S_i(t_j)}\right)\middle| S_i(t_j)\right] &= q(\ln u)^2 + (1-q)(\ln d)^2 - (q\ln u + (1-q)\ln d)^2 \\ &= \left[\ln\left(\frac{u}{d}\right)\right]^2 q(1-q). \end{aligned} \tag{2.18}$$

Note that the derivation of the binomial framework in this section is axiomatic and ignores the existence of a continuous-time market model. The above set of equations will be used to parameterize the binomial model in such a way that it becomes an approximation of the continuous time model established in the next section.

2.1.2 The Continuous-time Framework

To establish a continuous-time model for asset prices, the binomial assumption and discretized lattice are replaced with the following assumption, verbatim from Black and Scholes (1973):

> *The stock price follows a random walk in continuous time with a variance rate proportional to the square of the stock price. Thus the distribution of possible stock prices at the end of any finite interval is log-normal. The variance rate of the return on the stock is constant.*

Any stochastic differential equation for asset price movements that guarantees log-normality and constant variance of returns will satisfy the above and geometric Brownian motion is analytically tractable.

Consider the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with a filtration $\{\mathcal{F}_t, t \geq 0\}$ satisfying the usual conditions, where $t \in [0, T]$, and a standard Brownian motion W defined with respect to the filtration. Let $0 < t < s < T$ and propose

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t), \tag{2.19}$$

as a model for asset price movements, where μ is the annualized mean rate of return of the physical asset and σ is the annualized volatility of the rate of return. Applying Itô's Lemma yields

$$d\ln S(t) = \left(\mu - \frac{1}{2}\sigma^2\right) dt + \sigma dW(t). \tag{2.20}$$

The change to the risk-neutral measure $\mathbb{Q}$ is accomplished through the combined application of Girsanov's Theorem and the Martingale Representation Theorem[1], with the net result that the above two expressions become

$$dS(t) = rS(t)dt + \sigma S(t)d\widetilde{W}(t), \tag{2.21}$$

and

$$d\ln S(t) = \left(r - \frac{1}{2}\sigma^2\right) dt + \sigma d\widetilde{W}(t). \tag{2.22}$$

where $\widetilde{W}$ is a $\mathbb{Q}$-Brownian motion. The logarithmic form allows for an explicit solution for $S(s)$, given information until the current time t,

$$S(s) = S(t)e^{(r-\frac{1}{2}\sigma^2)(s-t)+\sigma(W_s-W_t)}. \tag{2.23}$$

Owing to the relationship $\ln S(s) - \ln S(t) = \ln \frac{S(s)}{S(t)}$, $d\ln S(t)$ represents an infinitesimal log-return whereas $\frac{dS(t)}{S(t)}$ is an infinitesimal rate of return. The model clearly makes no distinction between the annualized volatility of the rate of return of the asset and the annualized volatility of the log-return of the asset, which is assumed to be exogenous (an input to the model) and constant. Let

$$\alpha = r - \frac{1}{2}\sigma^2$$

[1] For a detailed primer, see Appendix B of Glasserman (2004)

be the annualized mean log-return under the risk-neutral measure. Then the expected return and expected log-return are respectively

$$\mathbb{E}^{\mathbb{Q}}\left[\frac{S(s)}{S(t)}\middle|\mathcal{F}_t\right] = e^{r(s-t)} \tag{2.24}$$

and

$$\mathbb{E}^{\mathbb{Q}}\left[\ln\frac{S(s)}{S(t)}\middle|\mathcal{F}_t\right] = \alpha(s-t). \tag{2.25}$$

The variance of asset price returns and log-returns are respectively

$$\mathbb{V}\mathrm{ar}^{\mathbb{Q}}\left[\frac{S(s)}{S(t)}\middle|\mathcal{F}_t\right] = e^{2r(s-t)}(e^{\sigma^2(s-t)}-1) \tag{2.26}$$

and

$$\mathbb{V}\mathrm{ar}^{\mathbb{Q}}\left[\ln\frac{S(s)}{S(t)}\middle|\mathcal{F}_t\right] = \sigma^2(s-t). \tag{2.27}$$

The above now form the continuous-time analogues of Equations (2.15), (2.16), (2.17) and (2.18). Note that the variance of the infinitesimal rate of return,

$$\mathbb{V}\mathrm{ar}\left[\frac{dS(t)}{S(t)}\middle|\mathcal{F}_t\right] = \sigma^2 dt,$$

is approximately equal to the variance of an infinitesimal return,

$$\mathbb{V}\mathrm{ar}^{\mathbb{Q}}\left[\frac{S(t+dt)}{S(t)}\middle|\mathcal{F}_t\right] = e^{2rdt}(e^{\sigma^2 dt}-1) \tag{2.28}$$

$$= \sigma^2 dt + O((dt)^2). \tag{2.29}$$

2.1.3 Parameterizations

A risk-neutral binomial tree is completely specified by the triplet of q, u, and d (with r exogenous) and thus requires three constraint equations. As seen in Section 2.1.1, enforcing no-arbitrage in the binomial framework results in an equation for q given by

$$q = \frac{e^{r\Delta t}-d}{u-d},$$

in Equation (2.7). Simple arithmetic shows that this is equivalent to matching the discrete-time mean return, Equation (2.15), to the continuous-time mean return, Equation (2.24), under the risk-neutral measure,

$$\mathbb{E}^{\mathbb{Q}}\left[\left.\frac{S(t_j+\Delta t)}{S_i(t_j)}\right| S_i(t_j)\right] = qu + (1-q)d = e^{r\Delta t}. \tag{2.30}$$

It remains to match the volatility to that of the continuous-time model. The choice between Equation (2.26) and Equation (2.27) will result in different parameterizations, but either condition will result in a correct approximation to the continuous log-normal distribution and return the correct volatility. Traditionally, the annualized volatility of log-returns is computed from the data and Equation (2.27) is used to provide

$$\mathbb{V}\mathrm{ar}^{\mathbb{Q}}\left[\left.\ln\left(\frac{S(t_j+\Delta t)}{S_i(t_j)}\right)\right| S_i(t_j)\right] = \left[\ln\left(\frac{u}{d}\right)\right]^2 q(1-q) = \sigma^2\Delta t. \tag{2.31}$$

Equations (2.30) and (2.31) are not sufficient to completely specify the model and thus multiple parameterizations exist.

Chance (2007) examines 11 well-established parameter specifications and finds that several do not require Equations (2.30) and (2.31) to hold and thus either allow arbitrage or incorrectly match volatility for a finite number of time steps (including the original CRR model from Cox *et al.* (1979) and the equally well-known Jarrow-Rudd model from Jarrow and Rudd (1983)). Although interesting, this is not truly a problem in practice as all the examined models correctly converge to the Black-Scholes-Merton model as $N \to \infty$ and N is usually large.

In fact, Hsia (1983) shows that if u and d are chosen to return the correct mean and volatility, any value of $q \in (0,1)$ will result in a binomial tree that will converge to the Black-Scholes-Merton model as $N \to \infty$. Chance (2007) uses this result to construct a general binomial parameterization with

$$u = \frac{e^{r\Delta t + \frac{\sigma\sqrt{\Delta t}}{\sqrt{q(1-q)}}}}{qe^{\frac{\sigma\sqrt{\Delta t}}{\sqrt{q(1-q)}}} + (1-q)}, \tag{2.32}$$

$$d = \frac{e^{r\Delta t}}{qe^{\frac{\sigma\sqrt{\Delta t}}{\sqrt{q(1-q)}}} + (1-q)} \tag{2.33}$$

and

$$q = \frac{e^{r\Delta t} - d}{u - d}. \tag{2.34}$$

In the above parameterization, q is assumed known and can be chosen as any value between 0 and 1. The corresponding construction of u and d will ensure that the tree returns the correct volatility and is arbitrage-free for any number of time-steps.

It is worthwhile to note that the multiplicative binomial tree described in this section will be recombining for any rational values of u and d. The additional condition that $ud = 1$ imposed by the CRR model is not necessary for recombination, instead it will ensure that for any even time step $2j\Delta t$, with $j = 0, 1, \cdots, \lfloor \frac{N}{2} \rfloor$, $S_{j+1}(2j\Delta t) = S(0)$.

In Joshi (2007), it was shown that the best performing binomial tree for pricing an American put option is one proposed by Tian (1993). The parameterization of this model is discussed in the next section.

Remark 2.1.2 – A Note on CRR and JR.
The attentive reader will immediately wonder why the original CRR model returns the incorrect volatility and why the Jarrow-Rudd model is not risk-neutral (both for a finite number of time steps only). This brief discussion is adapted from Chance (2007).

In Cox *et al.* (1979), the $\mathbb{P}$-measure equivalents of Equations (2.25) and (2.31) along with the additional constraint $ud = 1$ are used to completely solve for u, d and the physical probability p. This results in the parameterization

$$u = e^{\sigma\sqrt{\Delta t}}, \tag{2.35}$$
$$d = e^{-\sigma\sqrt{\Delta t}}, \tag{2.36}$$

and

$$p = \frac{1}{2} + \frac{1}{2}\frac{\mu}{\sigma}\sqrt{\Delta t}. \tag{2.37}$$

They ensure that their solution converges to the correct variance as $N \to \infty$ and then discard p in favour of the risk-neutral q from Equation (2.7). However, since u and d were determined from Equation (2.25) and not Equation (2.30), the solution only returns the correct volatility in the limit.

In Jarrow and Turnbull (1996), the $\mathbb{P}$-measure equivalents of Equations (2.25) and (2.31) are again used but p is set to $\frac{1}{2}$. This results in the parameterization

$$u = e^{\alpha\Delta t + \sigma\sqrt{\Delta t}}, \tag{2.38}$$
$$d = e^{\alpha\Delta t - \sigma\sqrt{\Delta t}}, \tag{2.39}$$

and

$$p = \frac{1}{2}. \tag{2.40}$$

To transform to $\mathbb{Q}$, α becomes $r - \frac{1}{2}\sigma^2$ and they use Equation (2.7) to show that q goes to $\frac{1}{2}$ in the limit. However, if q is set as $\frac{1}{2}$ the tree is not risk-neutral and if the above u and d are used, the correct volatility will be returned only in the limit.

The Tian3 Model

In Tian (1993) a binomial tree parameterization is proposed where the third-order non-central moment of the tree is matched to the third-order non-central moment of the log-normal distribution (see Remark 2.2.1). Using q_u and q_d as the risk-neutral up- and down-movement probabilities, the constraint equations for the model are

$$q_u + q_d = 1, \tag{2.41}$$
$$q_u u + q_d d = M, \tag{2.42}$$
$$q_u u^2 + q_d d^2 = M^2 W, \tag{2.43}$$

and

$$q_u u^3 + q_d d^3 = M^3 W^3, \tag{2.44}$$

where $M = e^{r\Delta t}$ and $W = e^{\sigma^2 \Delta t}$. Clearly, Equation (2.42) is equivalent to Equation (2.30) and thus the model will be free of arbitrage for any number of time-steps. Equation (2.43) correctly matches the volatility (although not in the traditional log-returns space) and thus the model will also return the correct volatility for any number of time-steps. Simultaneously solving the constraint equations presented will result in the parameterization known here as the Tian3 model, given by

$$q_u = \frac{M - d}{u - d}, \tag{2.45}$$
$$q_d = 1 - q_u = \frac{u - M}{u - d}, \tag{2.46}$$
$$u = \frac{MW}{2}\left[W + 1 + \sqrt{W^2 + 2W - 3}\right] \tag{2.47}$$

and

$$d = \frac{MW}{2}\left[W + 1 - \sqrt{W^2 + 2W - 3}\right]. \tag{2.48}$$

2.2 Trinomial Trees

2.2.1 The Trinomial Framework

The trinomial model is a straight-forward extension of the binomial model, with the binomial assumption replaced by the assumption that the asset price can attain one of three possible values over any small discrete time step. This can be written as

$$S(t_j+\Delta t) = \begin{cases} S_i(t_j+\Delta t) = uS_i(t_j) & \text{with risk-neutral probability } q_u, \\ S_{i+1}(t_j+\Delta t) = mS_i(t_j) & \text{with risk-neutral probability } q_m, \\ S_{i+2}(t_j+\Delta t) = dS_i(t_j) & \text{with risk-neutral probability } q_d, \end{cases} \tag{2.49}$$

under the risk-neutral measure $\mathbb{Q}$.

The relevant probabilities must sum to one and the model is constrained to match the first two non-central moments of the continuous-time distribution. This provides

$$q_u + q_m + q_d = 1, \tag{2.50}$$

$$\mathbb{E}^{\mathbb{Q}}\left[\frac{S(t_j+\Delta t)}{S_i(t_j)}\middle| S_i(t_j)\right] = q_u u + q_m m + q_d d = e^{r\Delta t}, \qquad \text{and} \tag{2.51}$$

$$\mathbb{E}^{\mathbb{Q}}\left[(\frac{S(t_j+\Delta t)}{S_i(t_j)})^2\middle| S_i(t_j)\right] = q_u u^2 + q_m m^2 + q_d d^2 = e^{(2r+\sigma^2)\Delta t}. \tag{2.52}$$

In the binomial model, a recombining tree was ensured for all rational values of u and d. To ensure recombination in the trinomial model, it is necessary to insist that

$$ud = m^2,$$

i.e. an up-move followed by a down-move is equal to two middle-moves. Insisting upon recombination reduces the order of the number of nodes in the tree from $\frac{3^{N+1}}{2}$ to $(N+1)^2$.

Currently there are 6 unknown parameters and 4 constraint equations and thus 2 additional restrictions are required to completely specify the model. This grants the trinomial tree an additional degree of freedom over the binomial tree and allows for more flexibility in the parameterization.

Remark 2.2.1 – A Note on Moments.
When matching the higher-order moments of the binomial or trinomial tree to the continuous-time distribution it is convenient to use the non-central moments (as opposed to the variance for the second-order moment), as the expressions for the non-central moments are immediately apparent in the discrete-time case.

For the trinomial model,

$$\mathbb{E}^{\mathbb{Q}}\left[\left(\frac{S(t_j+\Delta t)}{S_i(t_j)}\right)^n \middle| S_i(t_j)\right] = q_u u^n + q_m m^n + q_d d^n,$$

with the variables as defined in Section 2.2.1 and for the binomial model from Section 2.1.1,

$$\mathbb{E}^{\mathbb{Q}}\left[\left(\frac{S(t_j+\Delta t)}{S_i(t_j)}\right)^n \middle| S_i(t_j)\right] = q_u u^n + q_d d^n.$$

The n-th non-central moment of the continuous-time asset price process described in Section 2.1.2 is

$$\mathbb{E}^{\mathbb{Q}}[S(s)^n|\mathcal{F}_t] = S(t)^n e^{(nr+n(n-1)\frac{1}{2}\sigma^2)(s-t)}.$$

2.2.2 Parameterizations

As for the binomial model, a variety of trinomial trees are used in practice. Of interest here is the Tian4 parameterization proposed by Tian (1993), which was shown to be the best performing trinomial model (when acceleration techniques are used) for pricing an American put in Chan *et al.* (2009).

The Tian4 Model

The Tian4 tree is constructed by additionally matching the third and fourth order non-central moments of the tree, an idea which has intuitive appeal. Using the definitions

$$M = e^{r\Delta t} \tag{2.53}$$

and

$$W = e^{\sigma^2\Delta t}, \tag{2.54}$$

established in Section 2.1.3, matching the third- and fourth-order moments yield the two additional constraints

$$q_u u^3 + q_m m^3 + q_d d^3 = M^3 W^3$$

and

$$q_u u^4 + q_m m^4 + q_d d^4 = M^4 W^6.$$

The system is now completely specified and solving for the parameters yields

$$q_u = \frac{md - M(m+d) + M^2 W}{(u-d)(u-m)}, \tag{2.55}$$

$$q_m = \frac{M(u+d) - ud - M^2 W}{(u-m)(m-d)}, \tag{2.56}$$

$$q_d = \frac{um - M(u+m) + M^2 W}{(u-d)(m-d)}, \tag{2.57}$$

$$u = \kappa + \sqrt{\kappa^2 - m^2}, \tag{2.58}$$

$$d = \kappa - \sqrt{\kappa^2 - m^2}, \tag{2.59}$$

and

$$m = MW^2, \tag{2.60}$$

with $\kappa = \frac{M}{2}(W^4 + W^3)$.

In Tian (1993), the author states that the original trinomial model specified by Boyle (1986) will fail to converge when the volatility goes to zero, that is as $\sigma \to 0$. Although it is debateable whether this is a problem in practice, the Tian4 tree does not possess this potential difficulty.

Chapter 3

Errors in Tree Methods

We find that the best choice of tree depends on how one defines error...
— Mark S. Joshi (Joshi, 2007)

3.1 Classification of Errors

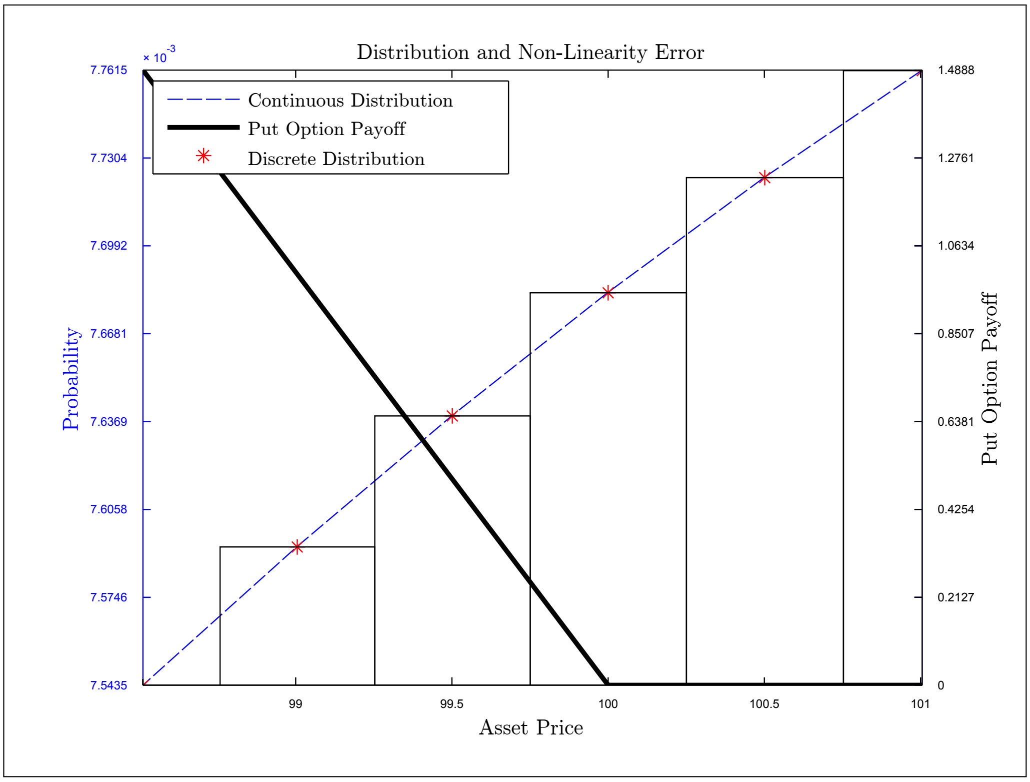
Before examining the techniques commonly used to accelerate the convergence of trees, or the development of more advanced tree methods, it is necessary to understand potential sources of error when using lattice methods in general and tree methods specifically. This exploration is constrained to four sources or classifications of error:

1. **Quantization Error**
   Quantization error arises from the unavoidably discrete nature of the lattice. A lattice generates an asset model that can only attain finite, discretized values and can only be observed at finite, discretized times. The option price generated by a lattice method will theoretically be correct only for an option written on an underlying that exhibits this unrealistic discrete behaviour.

2. **Option Specification Error**
   This type of error occurs when a lattice fails to correctly capture the contractual terms of the option. An illustrative example is that of a barrier option, when there is not a layer of nodes in the lattice coinciding with the barrier. The lattice will then value an option where the effective barrier is actually above or below the true barrier specified in the contract.

   A similar situation could occur when valuing an option with Bermudan-style exercise. Should the discretized time steps not coincide with the correct possible exercise dates, option specification error will be present in the valuation.

Fig. 3.1: Distribution and non-linearity error around the strike at expiry for a European put, modelled after Figure 1 in Figlewski and Gao (1999). The parameters are $S(0) = 100,\ K = 100,\ r = 0.1,\ \sigma = 0.25$ and $T = 1$.



3. **Distribution Error**

   Distribution error arises from approximating the continuous log-normal distribution with a discrete distribution (the binomial distribution in the case of binomial trees). Consider the node at $S(T) = 99.5$ in Figure 3.1. In the lattice the probability of the node is constant across the interval, however in the figure it is clear that the true probability varies quite sharply across the interval.

   The difference between weighting the mean option payoff at the node with the single discrete probability as opposed to the continuous probability gives rise to distribution error.

4. **Non-linearity Error**

   Non-linearity error occurs where the option value function is highly non-linear. The option value at a node represents the option value across the interval the node covers. However, when the option is highly non-linear a single point in the interval is a very poor approximation of the option value. This is what occurs at the strike, $S(T) = 100$, in Figure 3.1. In the lattice, the option value for that node is zero, whereas it is clear that the option truly has a non-zero value across the interval.

The classifications of quantization error and option specification error were identified in Derman *et al.* (1995) and apply to all lattice methods, whereas distribution error and non-linearity error were highlighted in Figlewski and Gao (1999) and were presented as specific to tree methods.

It should be clear that errors 2 through 4 are truly subcategories of the first listed: quantization error. If quantization error could be eliminated, the model would be continuous and exact (or at least an exact representation of the continuous model on which it is based). If quantization error is reduced, the effect of the other error classifications will also decrease, although in differing proportions. The subcategories of error are useful however, because they can be used to explain the different convergence rates of different methods and it is possible to reduce them separately from quantization error. Specifically, Derman *et al.* (1995) provides a method for modifying binomial trees to reduce the potential option specification error when valuing barrier options. This is explored in Section 6.2.1.

The adaptive mesh model was designed to greatly reduce the effect of non-linearity error and this will be seen in Section 5.2.

Remark 3.1.1 – A Note on Linearity.
As an aside, a linear, path-independent option (such as a forward) will be perfectly valued by a correctly constructed lattice method as shown in Figlewski and Gao (1999):

$$\begin{aligned} V_{\text{Lattice}}(0) &= e^{-rT}\mathbb{E}^{\mathbb{Q}}_{\text{Lattice}}[V(S(T))] = e^{-rT}V(\mathbb{E}^{\mathbb{Q}}_{\text{Lattice}}[S(T)]) \\ &= e^{-rT}V(\mathbb{E}^{\mathbb{Q}}[S(T)]) = V(0) \end{aligned}$$

Since the payoff of the option is linear, it can be taken out of the expectation. By construction, the expected value of the asset in the lattice will match the true risk-neutral expected value.

Chapter 4

Acceleration Techniques

> *We find that the best choice of trinomial tree depends on how one defines error, but in all cases one should use the acceleration techniques of smoothing, Richardson extrapolation and truncation.*
>
> — (Chan *et al.*, 2009)

Four different techniques are commonly used to accelerate the convergence of both binomial and trinomial trees. They are discussed here in turn, and, barring the control variate technique, implemented to accelerate the chosen traditional tree methods.

4.1 Smoothing

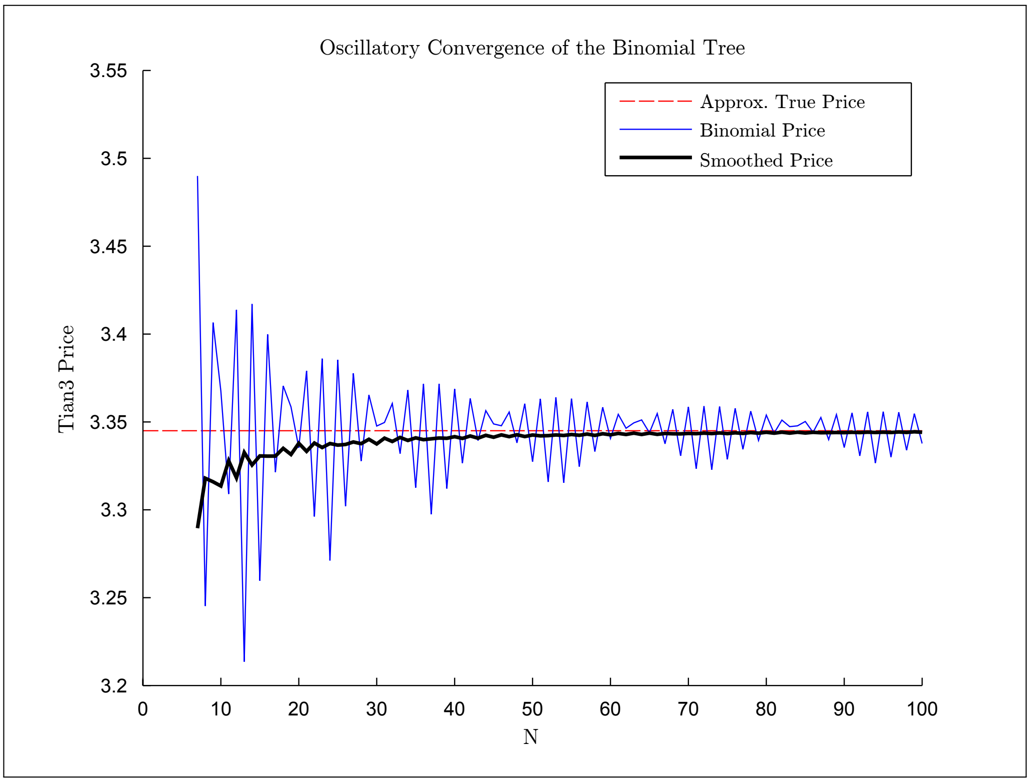
It is well-established (at least for vanilla options) that binomial trees have $\mathcal{O}(\frac{1}{N})$ convergence (see Leisen (1998) for the American put option case), and that the lead error term is oscillatory in nature[1].

The technique of smoothing, proposed by Broadie and Detemple (1996), is conceptually very simple: the continuation value of the option at the time step just before maturity is replaced with the Black-Scholes option value. In the discrete time framework, the last exercise opportunity occurs at time $t_{N-1} = (N-1)\Delta t$. This implies that every option has European-style exercise over the last time interval. Thus the value of the option at this time in the tree is the Black-Scholes-Merton price for an European option created at time t_{N-1} with expiry t_N.

The smoothing technique has the effect of dampening the oscillations in the convergence of the binomial tree, as illustrated in Figure 4.1.

[1] The lead error term is driven by the distance in log-space between the strike and the neighbouring nodes. For a complete treatment of the oscillatory convergence, see Diener and Diener (2004) and Walsh (2003).

Fig. 4.1: The convergence of the Tian3 model with and without smoothing for an American put option. The parameters are $S(0) = 100$, $K = 90$, $r = 0.05$, $\sigma = 0.30$ and $T = 0.5$. The true price is taken as 3.345, as per the example in Broadie and Detemple (1996).



4.2 Richardson Extrapolation

Although the technique is also due to Broadie and Detemple (1996), the explanation in this section is derived from Chen and Joshi (2012).

Let V_N be the price of the option generated by a tree with N time steps, V_N^{RE} be the price with Richardson extrapolation applied and V_{True} be the correct price. If it is assumed that the option price generated by the N-step tree can be written as

$$V_N = V_{\text{True}} + \frac{\epsilon}{N} + \mathcal{O}\left(\frac{1}{N^2}\right), \tag{4.1}$$

with ϵ a constant[2], then Richardson extrapolation can be used to eliminate the $\frac{\epsilon}{N}$ term, resulting in

$$V_N^{\text{RE}} = V_{\text{True}} + \mathcal{O}\left(\frac{1}{N^2}\right). \tag{4.2}$$

[2] For a stronger result using little-$\mathcal{O}$ notation, see Chen and Joshi (2012).

To accomplish this, a weighted sum is taken of the N-step tree price and a price generated by a tree with $\lfloor \frac{N}{2} \rfloor$ steps,

$$V_N^{\mathrm{RE}} = wV_N + (1-w)V_{\lfloor \frac{N}{2} \rfloor}, \tag{4.3}$$

which provides the constraint for w as

$$w\frac{\epsilon}{N} + (1-w)\frac{\epsilon}{\lfloor \frac{N}{2} \rfloor} = 0. \tag{4.4}$$

Solving the above yields

$$w = \begin{cases} 2 & \text{if } N \text{ is even}, \\ \frac{2N}{N+1} & \text{if } N \text{ is odd}, \end{cases} \tag{4.5}$$

which for even N gives the simple formula

$$V_N^{\mathrm{RE}} = 2V_N - V_{\lfloor \frac{N}{2} \rfloor}. \tag{4.6}$$

Now, Equation (4.1) is not generally true for American put options, as the leading error term is oscillatory, but applying the smoothing technique from the previous section leads to convergence of such a nature that Richardson extrapolation can be used successfully.

4.3 Truncation

Established by Andricopoulos *et al.* (2004), the truncation method proposes using only nodes within ξ standard deviations in log-space from the risk-neutral mean of the asset price or within ξ standard deviations of the present value of the strike, to value the option. Typically, $5 \leq \xi \leq 7$.

This provides the upper and lower asset price bounds at time step j as

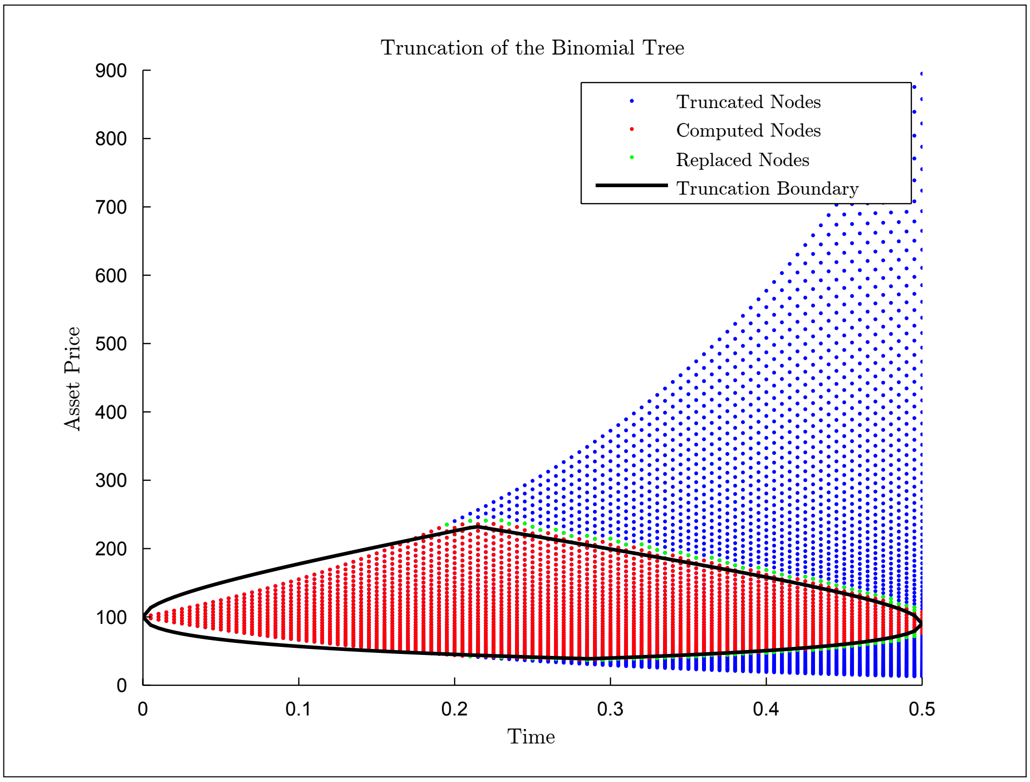
$$S_{\max}(t_j) = \min(S(0)e^{rt_j+\xi\sigma\sqrt{t_j}}, Xe^{-r(T-t_j)+\xi\sigma\sqrt{T-t_j}}) \tag{4.7}$$

and

$$S_{\min}(t_j) = \max(S(0)e^{rt_j-\xi\sigma\sqrt{t_j}}, Xe^{-r(T-t_j)-\xi\sigma\sqrt{T-t_j}}) \tag{4.8}$$

and these boundaries are illustrated in Figure 4.2. The above two equations are then used in conjunction with the equations for the stock price at any time in the

Fig. 4.2: The effect of truncation on the Tian3 binomial tree. The parameters are as in Figure 4.1, with $N = 100$ and truncation reduced the node count by 44.6 %.



tree to establish the maximum and minimum nodes that should be calculated.

Consider the equation for the asset price at node i and time t_j in a binomial tree,

$$S_i(t_j) = u^{j-(i-1)} d^{i-1} S(0), \tag{4.9}$$

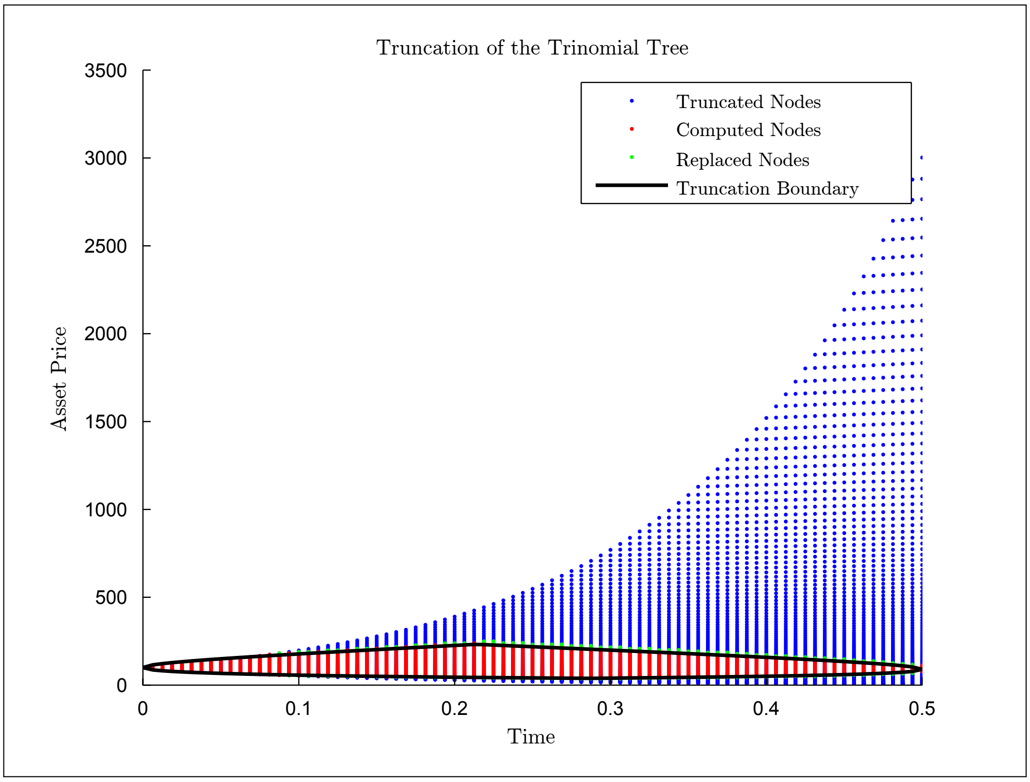
where $1 \leq i \leq j+1$ and $0 \leq j \leq N$. Inverting this equation for i and substituting the expressions for $S_{\max}(t_j)$ and $S_{\min}(t_j)$ yields

$$i_{\max}(t_j) = \left(\frac{\ln \frac{S_{\max}(t_j)}{S(0)u^j})}{\ln \frac{d}{u}} \right) + 1 \tag{4.10}$$

and

$$i_{\min}(t_j) = \left(\frac{\ln \frac{S_{\min}(t_j)}{S(0)u^j})}{\ln \frac{d}{u}} \right) + 1, \tag{4.11}$$

Fig. 4.3: The asset price nodes generated by the Tian4 trinomial tree, as well as the nodes remaining after truncation. Created with $N = 80$ and the same option parameters as in Figure 4.1, the node count was reduced by 60.7 %.



where $i_{\max}(t_j)$ and $i_{\min}(t_j)$ will be the numbers of the highest and lowest nodes to include in the computation at time t_j. Similar equations are easily derived for a trinomial tree.

Care must be taken when implementing Equations (4.10) and (4.11), since $S_{\max}(t_j)$ and $S_{\min}(t_j)$ are not constrained by Equation (4.9) and thus $i_{\max}(t_j)$ and $i_{\min}(t_j)$ will not necessarily be integers. Appropriate rounding must be applied.

The calculation of the value of the option at a parent node can require knowledge of a child node that has been truncated. In this case, a replacement value is used for the value of the option at the child node. Andricopoulos *et al.* (2004) uses the intrinsic value of the option (the value upon immediate exercise), and this is used in this work. However, for further discussion regarding potential replacement values see Chen and Joshi (2012).

4.4 Control Variates

The control variate technique was established by Hull and White (1988) and is specific to American options. It is based on the principle that the size of the error present in a tree price for an American option will be related to the size of the error when the same tree is used to price a European option, for which the correct price is known.

Let V_N be the price of the American option generated by a tree with N time steps, V_N^E be the price generated by the same tree for the European version of the option and V^{BS} be the Black-Scholes price for the European option. The price generated by the control variate technique, V^{CV}, is given by

$$V^{\text{CV}} = V_N + (V^{\text{BS}} - V_N^E). \tag{4.12}$$

In Joshi (2007) it was shown that the control variate technique has been superseded by Richardson extrapolation implemented in conjunction with smoothing, at least when pricing American put options. As such, it is not implemented in this work.

Chapter 5

The Adaptive Mesh Model

A straight-forward way to reduce non-linearity error (illustrated in Figure 3.1) is to increase the resolution of the tree by increasing the number of time-steps, N. This has the effect of increasing the number of nodes in the non-linear region, thus forming a better approximation of the true option value. However, computational effort will then be wasted in regions where the option value is linear and a higher resolution is not needed. The Adaptive Mesh Model (AMM) provides a mechanism to vary the resolution of the tree in small sections where greater accuracy is required. It does this by grafting one or more refined lattices onto the course tree.

The AMM model was developed in Figlewski and Gao (1999) and is based on an arithmetic trinomial tree structure which computes asset values in log-space. The parameterization of this tree will be referred to as the Gao1 parameterization, as opposed to the AMM model which includes the mesh refinement.

5.1 The Gao1 Parameterization

Consider the log of the asset price in continuous time under the risk-neutral measure,

$$d \ln S(t) = \alpha dt + \sigma dW_t, \tag{5.1}$$

with $\alpha = r - \frac{1}{2}\sigma^2$, as defined in Section 2.1.2. Let

$$X(t) = \ln S(t) - \alpha t \tag{5.2}$$

be the mean adjusted value of the log of the asset price. A change in $X(t)$ over an increment of time will be distributed normally and centered around 0, which can be seen from

$$X(t+\Delta t) - X(t) = \ln \frac{S(t+\Delta t)}{S(t)} - \alpha \Delta t \tag{5.3}$$

$$\sim \mathcal{N}(0, \sigma^2 \Delta t). \tag{5.4}$$

The Gao1 trinomial tree models the movement of $X(t_j)$ over the discrete times $t_j = j\Delta t$, with $j = 0, \cdots, N-1$, as

$$X(t_j + \Delta t) = \begin{cases} X_i(t_j + \Delta t) = X_i(t_j) + h & \text{with probability } q_u, \\ X_{i+1}(t_j + \Delta t) = X_i(t_j) & \text{with probability } q_m, \\ X_{i+2}(t_j + \Delta t) = X_i(t_j) - h & \text{with probability } q_d, \end{cases} \tag{5.5}$$

where h is the size of an arithmetic up or down movement. It should be clear that this tree is symmetric around $X(0) = \ln S(0)$.

The tree will be completely parameterized by the specification of $q_u,\ q_m,\ q_d$ and h. Figlewski and Gao (1999) use the mean, the first two even non-central moments of the Normal distribution and the condition that the probabilities must sum to one to provide four constraint equations,

$$q_u + q_m + q_d = 1, \tag{5.6}$$

$$\mathbb{E}^{\mathbb{Q}}[X(t_j + \Delta t) - X(t_j) | X(t_j)] = p_u h + p_m 0 + p_d(-h) = 0, \tag{5.7}$$

$$\mathbb{E}^{\mathbb{Q}}[(X(t_j + \Delta t) - X(t_j))^2 | X(t_j)] = p_u h^2 + p_m^2 0 + p_d h^2 = \sigma^2 \Delta t, \tag{5.8}$$

and

$$\mathbb{E}^{\mathbb{Q}}[(X(t_j + \Delta t) - X(t_j))^4 | X(t_j)] = p_u h^4 + p_m 0 + p_d h^4 = 3\sigma^4 \Delta t^2. \tag{5.9}$$

Solving the above four equations simultaneously provides the parameterization

$$q_u = \frac{1}{6},\ q_m = \frac{2}{3},\ q_d = \frac{1}{6}$$

and

$$h = \sigma\sqrt{3\Delta t}. \tag{5.10}$$

Remark 5.1.1 – A Note on Gao1.
The Gao1 model is a very specifically constructed parameterization and it is clear from the selected constraint equations that the resulting tree will not be risk-neutral, since

$$\mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t}S(t_j+\Delta t)|S(t_j)] = S(t_j)e^{-r\Delta t}\left[\frac{1}{6}e^{\sigma\sqrt{3\Delta t}}+\frac{2}{3}+\frac{1}{6}e^{-\sigma\sqrt{3\Delta t}}\right] \tag{5.11}$$

$$\neq S(t_j). \tag{5.12}$$

The measure will still be denoted by $\mathbb{Q}$ as the tree was parameterized with reference to the risk-neutral measure in continuous time.

5.2 The AMM Model

The AMM model is a Gao1 trinomial tree with a higher resolution mesh, using the same parameterization, grafted onto it in the region where the option value is highly non-linear. In the language of hedging, this would occur in the price region where the option has a high gamma.

In this section, an American put is considered, where the non-linearity occurs around the strike at expiry. An overview of how the AMM is adapted to value a barrier option is given in Section 6.2.2.

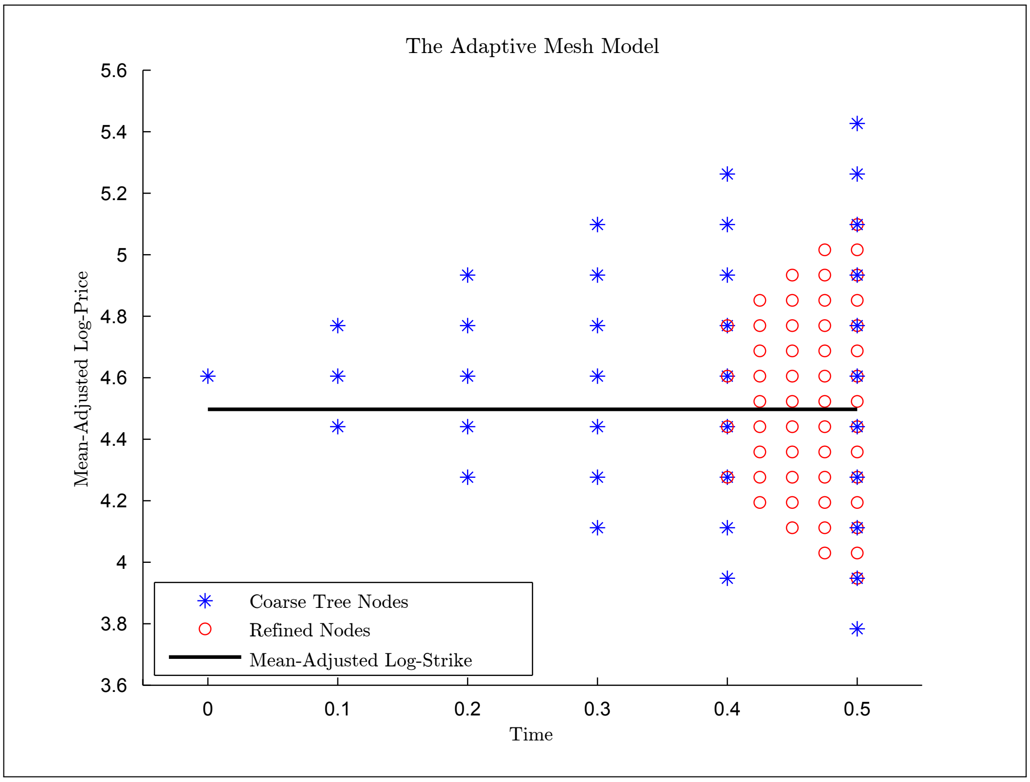
The adaptive mesh model is illustrated in Figure 5.1. The option is non-linear around the strike, K, which in the Gao1 parameterization transforms to $X^K(T) = \ln K - \alpha T$, depicted by the black line. It is around this line, at $t_N = T$, that a higher resolution lattice is required.

To correctly join the higher resolution lattice to the coarse tree, the nodes in the fine mesh must overlap nodes in the coarse tree at t_{N-1}. Without this condition, valuation information would not be transmitted properly. In order to achieve this, the arithmetic step-size for the fine mesh is set to $\frac{h}{2}$ where h is the step-size in the coarse tree. This leads to the discrete time-intervals in the fine-mesh being of size $\frac{\Delta t}{4}$.

At t_{N-1} the option value at the nodes in the fine mesh will replace the option value at the nodes they overlap in the coarse mesh. This is illustrated in Figure 5.1 at $t = 0.4$.

Non-linearity error in the tree will be reduced by overwriting all coarse nodes at t_{N-1} from which there are fine mesh paths that end up both in and out of the money.

Fig. 5.1: A 5-step adaptive mesh model for an American put option. The parameters are $S(0) = 100,\ K = 90,\ r = 0.05,\ \sigma = 0.30$ and $T = 0.5$ as in Figure 4.1.



Let $X^{K^+}(t_N)$ be the node directly above $X^K(T)$ and $X^{K^-}(t_N)$ be the node directly below. The coarse nodes at t_{N-1} that are to be overwritten will be bracketed below by $X^{K^+}(t_N) - 2h$ and above by $X^{K^-}(t_N) + 2h$. Therefore if $X^{K^+}(t_N) = X_i(t_N)$, the "origin" nodes for the higher resolution lattice will range from $n_{i-2}(t_{N-1})$ to $n_{i+1}(t_{N-1})$.

Once the origin nodes have been identified, the procedure for generating the higher resolution lattice is identical to a normal Gao1 trinomial tree, except that for the first step the tree is not recombining. It is important to note that the AMM is isomorphic at successive levels of refinement, which eases the implementation.

Chapter 6

Pricing

Whenever risk neutral valuation is possible, any approximation procedure based on a probability distribution that approximates the risk neutral distribution and converges to it in the limit can be used to price options correctly.

— S. Figlewski, B. Gao (Figlewski and Gao, 1999)

6.1 Pricing an American Put

In this section, the accelerated traditional tree methods are compared to the more advanced Adaptive Mesh Model (AMM) when pricing an American put. The original Cox-Ross-Rubenstein tree is used as a baseline for comparison. The models are summarized in Table 6.1.

The analysis is executed according to the framework established in Broadie and

Tab. 6.1: Model summary for pricing an American Put

Model:	**Summary:**
CRR	The standard Cox-Ross-Rubinstein binomial tree.
Tian3A	The Tian3 binomial tree with smoothing, Richardson extrapolation and truncation with parameter $\xi = 6$.
Tian4A	The Tian4 trinomial tree with smoothing, Richardson extrapolation and truncation with parameter $\xi = 6$.
AMM1	The Adaptive Mesh Model with one layer of refinement around the strike at expiry.
AMM2	The Adaptive Mesh Model with two layers of refinement around the strike at expiry.

Detemple (1996). 2500 American put options are randomly generated with the following parameter distribution:

Parameter:	**Distribution:**
$S_1(t_0)$	Uniformly distributed on $(70, 130)$.
σ	Uniformly distributed on $(0.1, 0.6)$.
r	Uniformly distributed on $(0.0, 0.1)$ with probability 0.8 and equal to 0 with probability 0.2.
T	Uniformly distributed on $(0.1, 1.0)$ with probability 0.75 and uniformly distributed on $(1.0, 5.0)$ with probability 0.25.
K	Constant at 100.

A "true" price for each option is then computed using a 15 000 step CRR tree. Any convergent tree method could be used, but using the CRR tree is standard practice in the literature. Options with prices less than 0.5 are discarded from the sample, to avoid distortions caused by small errors present in small option values. The remaining options are then all priced using the selected models for a specified number of time-steps.

Two metrics are used to determine efficiency: root mean squared relative error (RMS) and options priced per second (OPS). Root mean squared relative error is defined as

$$\text{RMS} = \sqrt{\frac{1}{m}\sum_{i=1}^{m} c_i^2} \tag{6.1}$$

and

$$c_i = \frac{\hat{c}_i - c_i}{c_i}, \tag{6.2}$$

where $\hat{c}_i$ is the estimated option value given by the model and c_i is the accepted true option value given by the 15 000 step CRR tree.

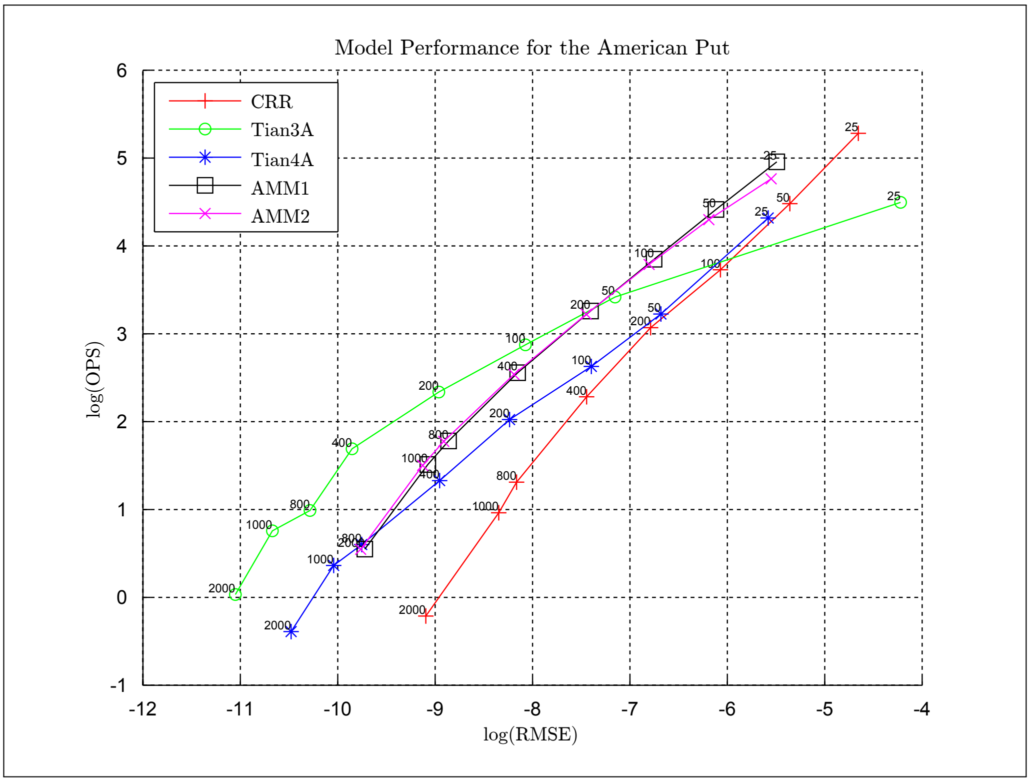
6.1.1 Results

Option pricing was performed for N-values of

$$N = 25, 50, 100, 200, 400, 800, 1000, 2000.$$

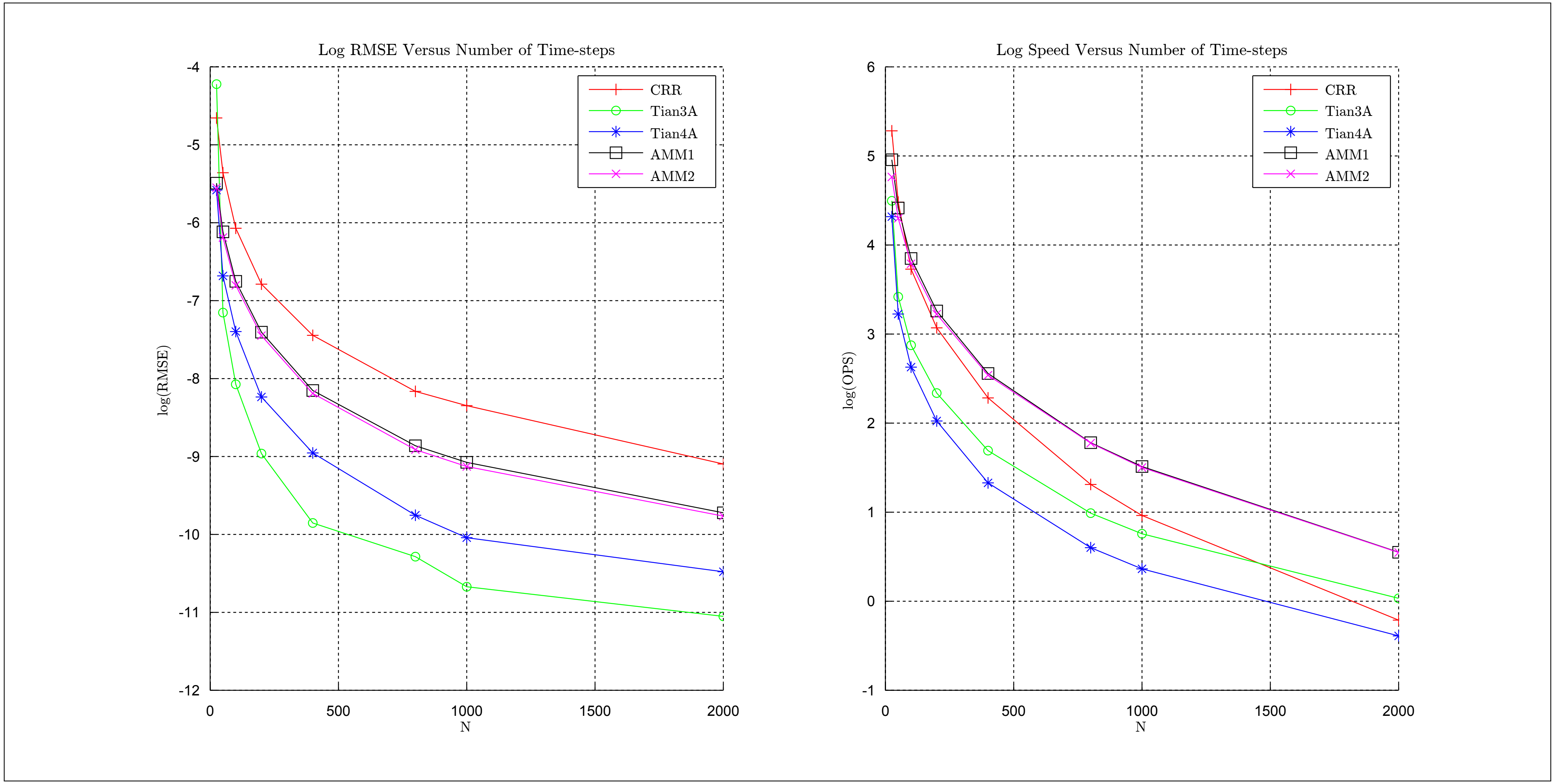
Note that for $N = 25$, Richardson extrapolation will not occur in the Tian3A and Tian4A trees. The resulting metrics are displayed in Tables 6.2 and 6.3. A log-plot

Fig. 6.1: The log-metrics for 2500 randomly generated American put options priced on the five specified models.



of the speed (measured in options per second) versus root mean squared error for all five models is given in Figure 6.1. The individual metrics are plotted against the values of N in Figure 6.2.

Fig. 6.2: The individual log-metrics plotted against the number of time-steps.



Tab. 6.2: Comparison across the five tree methods when pricing 2500 randomly generated American put options for the first four selected time-steps.

Model	**Time-steps**							
	25		50		100		200	
	RMS	OPS	RMS	OPS	RMS	OPS	RMS	OPS
CRR	9.5112e-03	196.7281	4.7085e-03	88.3592	2.3090e-03	41.6349	1.1271e-03	21.5376
Tian3A	1.4678e-02	89.6633	7.8246e-04	30.5204	3.1171e-04	17.7165	1.2816e-04	10.3536
Tian4A	3.7692e-03	75.0567	1.2534e-03	25.1572	6.1340e-04	13.8543	2.6495e-04	7.5675
AMM1	4.1181e-03	142.1905	2.2055e-03	82.7004	1.1685e-03	46.9265	6.0844e-04	26.0295
AMM2	3.8865e-03	117.2227	2.0501e-03	73.6254	1.1100e-03	44.2291	5.8230e-04	25.0953

Tab. 6.3: Comparison across the five tree methods when pricing 2500 randomly generated American put options for the remaining four selected time-steps.

Model	**Time-steps**							
	400		800		1000		2000	
	RMS	OPS	RMS	OPS	RMS	OPS	RMS	OPS
CRR	5.8438e-04	9.8067	2.8472e-04	3.7113	2.3707e-04	2.6193	1.1214e-04	0.8088
Tian3A	5.2593e-05	5.4254	3.4139e-05	2.6888	2.3205e-05	2.1331	1.5869e-05	1.0332
Tian4A	1.2922e-04	3.7771	5.8085e-05	1.8251	4.3541e-05	1.4383	2.8125e-05	0.6776
AMM1	2.8796e-04	12.8999	1.4167e-04	5.9321	1.1449e-04	4.5381	5.9938e-05	1.7330
AMM2	2.7732e-04	12.6543	1.3475e-04	5.9024	1.0854e-04	4.4914	5.7434e-05	1.7282

The first result to note is the confirmation of the findings established in Chan *et al.* (2009): the Tian3A tree dominates the Tian4A tree by being faster and more accurate for all $N \geq 50$. The CRR model, used here as a baseline, as the highest RMSE but is also the second fastest model until $N = 2000$, where the Tian3A model prices faster. This is possibly owing to the speed-up effects of the truncation technique being more pronounced at higher numbers of time-steps.

The AMM1 tree is faster and less accurate than the AMM2 tree when the N is small, but the difference between the methods becomes negligable for higher N-values. The AMM2, as expected, retains an advantage by having a lower RMSE.

The true comparison, then, is between the Tian3A model and the AMM1 and AMM2 models. As per Chan *et al.* (2009), linear interpolation is used on the log-speed and log-error to perform fair evaluation. The AMM1 root mean squared relative error is interpolated to yield the value corresponding to the same computation time as the Tian3A model. The results are shown in Figure 6.3, along with the interpolated approximate number of time-steps the AMM1 model would contain at the appropriate speed.

It is clear from the figure that the AMM1 model outperforms the Tian3A for very sparse trees, but as the density of the Tian3A tree is increased to $N > 50$, it dominates that corresponding AMM1 tree by yielding a lower root mean squared error for the same computation time. It remains to be seen how each model performs when pricing a more intricate option: the Down-And-Out put in the next section.

6.2 Pricing a European Barrier Option

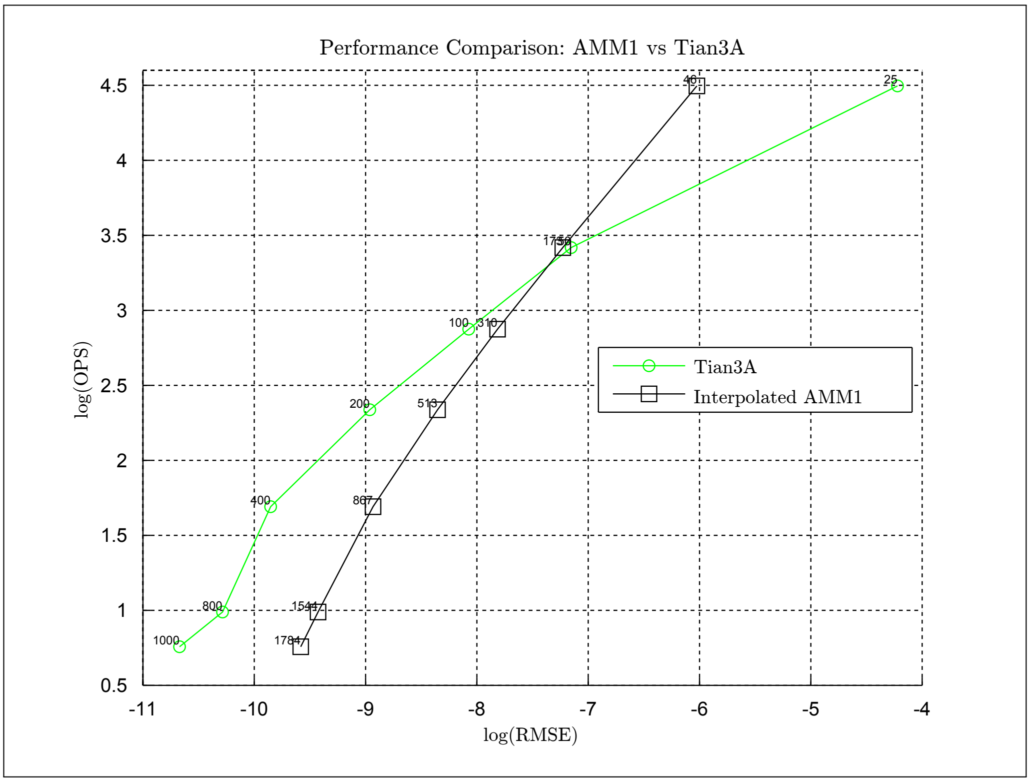
The difficulties of pricing barrier options when the initial asset price is close to the barrier are well established[1]. The origin of the problem lies with the number of asset price movements in the tree between initial price and the barrier.

As an example, consider a tree where there are two asset price steps between the initial asset price and a knock-out barrier. If the size of the time-step is decreased slightly, it is possible that it will now require three downward moves through the lattice for the option to be knocked out. Three downward movements is a significantly lower probability event than the two downward moves previously required. Thus a small decrease in the size of the time-steps (or a small increase in N) can produce a significant drop in the probability that the option is knocked out over its lifetime and thus a sizeable upward “jump” in its value (as computed on the lattice). This results in poor and jagged convergence when valuing barrier options using traditional tree methods (see Figure 6.6).

Two techniques are presented in this section to deal with this issue. The first is a modification of the traditional binomial tree method proposed by Derman *et al.* (1995) and designated here as the Modified Barrier Algorithm. The second is an alternate parameterization and implementation of the Adaptive Mesh Model (AMM) specifically to deal with barrier options.

[1] See Figlewski and Gao (1999) for a brief overview of modelling techniques to deal with this problem.

Fig. 6.3: Comparison of the Tian3A model versus the AMM1 at the same log-speed.



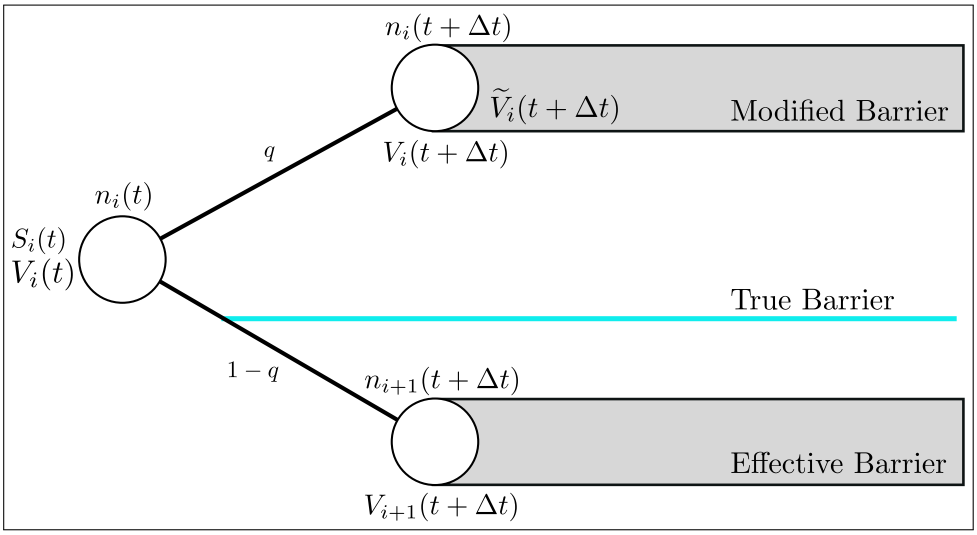
The efficiency of the two models will be compared when pricing a European Down-And-Out put option. The Down-And-Out put was selected as it has a very narrow band of non-zero option values in the lattice when the initial asset price is close to the barrier and thus is quite difficult to accurately price. European exercise was chosen as analytical solutions exist for European barrier options which allows for easy analysis of the error present in the models.

6.2.1 Modifying the Tian3A Model

Derman *et al.* (1995) proposed a technique for decreasing option specification error when pricing barrier options on a lattice. The work proposed two separate but equivalent interpretations of the technique: interpolation at the barrier or a Taylor expansion around the barrier. The Taylor expansion formulation is presented here.

Consider Figure 6.4, which illustrates the nodes in a binomial tree around the barrier of a Down-And-Out put option. The barrier specified in the option contract is known as the true barrier and is highlighted in cyan. The effective barrier is where the lattice first "feels" the effect of the barrier. This is the first layer of nodes in the tree where the option will be knocked out. When the option is valued at the effective barrier, the value, $V_i(t + \Delta t)$, at the node above the barrier will in actuality be too high, as it is not taking into account how close the node is to the true barrier. The

Fig. 6.4: An illustration of a group of nodes in a binomial tree clustered around the knock-out barrier of a Down-And-Out put option.



layer of nodes lying above the effective barrier will be known as the modified barrier. Note that similar to the effective barrier, the modified barrier is not a single asset value, but rather the value at each time step of the first node above the barrier.

Derman *et al.* (1995) propose pricing the option in the tree at the effective barrier and then using the resulting tree to update the modified barrier nodes with more accurate values, $\widetilde{V}_i(t+\Delta t)$. Finally, the option is priced again on the interior nodes of the tree, using the updated modified barrier values and resulting in a more acccurate price.

To update the modified barrier nodes, consider a Taylor expansion around the option value at the barrier,

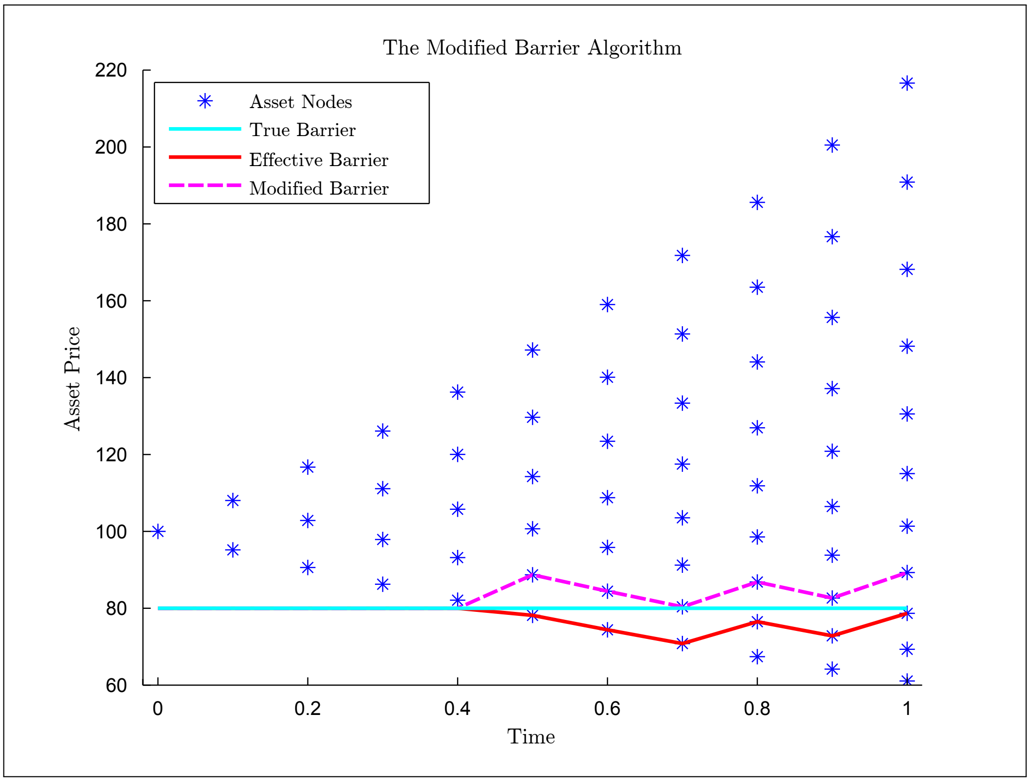
$$V(B) \approx V_i(t+\Delta t) + V'(B)(B - S_i(t+\Delta t)). \tag{6.3}$$

To approximate $V'(B)$, the option must first be valued on the original tree. This will allow for a finite-difference approximation to $V'(B)$ to be computed as

$$V'(B) \approx \frac{V_i(t+\Delta t) - V_{i+1}(t+\Delta t)}{S_i(t+\Delta t) - S_{i+1}(t+\Delta t)}. \tag{6.4}$$

Finally, $\widetilde{V}_i(t+\Delta t)$ replaces the original node at the modified barrier in the Taylor

Fig. 6.5: The Modified Barrier Algorithm for a European Down-And-Out put. The parameters are $S(0) = 100,\ K = 100,\ r = 0.1,\ \sigma = 0.2,\ T = 1$ and the barrier is set at 80.



expansion, yielding

$$\widetilde{V}_i(t+\Delta t) = V(B) - \frac{V_i(t+\Delta t) - V_{i+1}(t+\Delta t)}{S_i(t+\Delta t) - S_{i+1}(t+\Delta t)}(B - S_i(t+\Delta t) \tag{6.5}$$

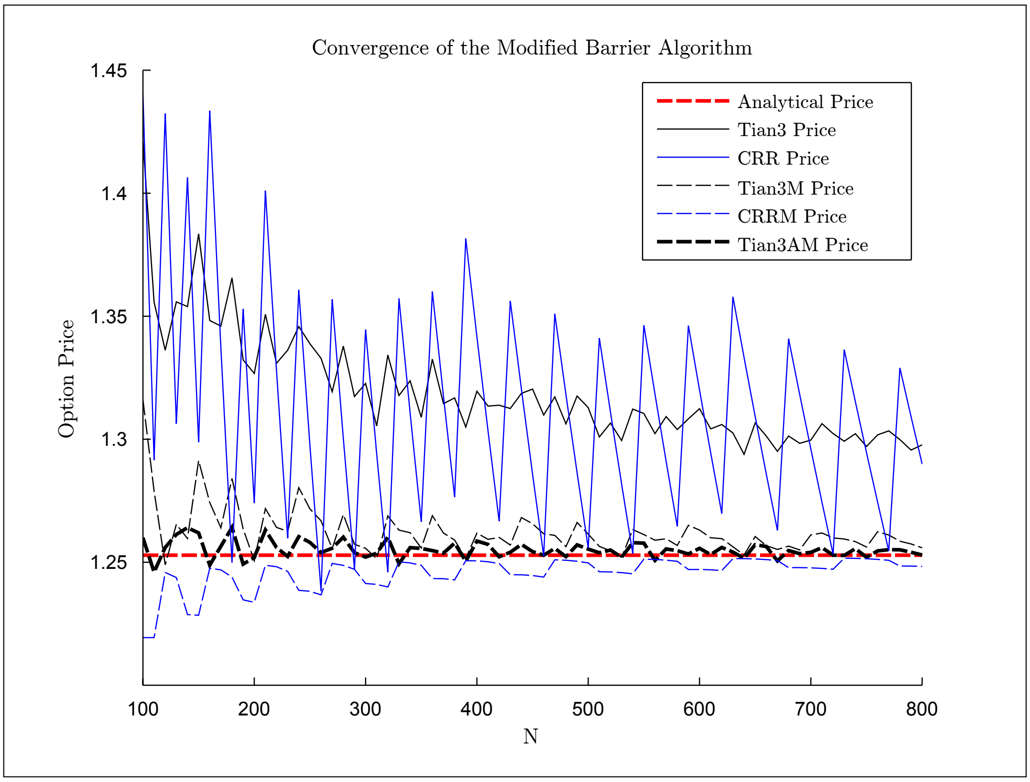
$$= -\frac{V_i(t+\Delta t)}{S_i(t+\Delta t) - S_{i+1}(t+\Delta t)}(B - S_i(t+\Delta t), \tag{6.6}$$

where $V(B) = 0$ for a knock-out barrier. The algorithm is illustrated in Figure 6.5 and the effect on convergence in Figure 6.6. Please note that the "M" designation has been appended to the models from Table 6.1 that have now been implemented with the Modified Barrier Algorithm.

6.2.2 Modifying the AMM Model

The Adaptive Mesh Model was created to reduce the effect of non-linearity error, which is exaggerated when pricing barrier options as the option value is non-linear everywhere the lattice crosses the barrier. To price barrier options, an alternate parameterization of the AMM is needed, designated here as the Gao2 parameteri-

Fig. 6.6: The convergence of the standard Tian3 and CRR trees compared to the Modified Barrier Algorithm. The parameters are as given in Figure 6.5.



zation. Note that the Gao2 parameterization cannot exist independently from the AMM, to use it effectively there must be at least one level of refined mesh attached. This contrasts with Gao1, which can be implemented independently from the AMM as a normal trinomial pricing model.

The Gao2 Parameterization

From Section 5.1, h is retained as the size of an arithmetic up or down movement in the tree, whereas the size of a middle move remains 0. To keep the distance between the barrier and initial node constant throughout the tree, the Gao2 parameterization does not adjust for the mean and instead only models the log of the asset price,

$$X(t) = \ln(S_t).$$

The constraints in the Gao1 parameterization, Equations 5.6 to 5.9, become

$$q_u + q_m + q_d = 1, \tag{6.7}$$

$$\mathbb{E}^{\mathbb{Q}}[X(t_j + \Delta t) - X(t_j)|X(t_j)] = p_u h + p_m 0 + p_d(-h) = \alpha \Delta t, \tag{6.8}$$

$$\mathbb{E}^{\mathbb{Q}}[(X(t_j + \Delta t) - X(t_j))^2|X(t_j)] = p_u h^2 + p_m^2 0 + p_d h^2 = \alpha^2 \Delta t^2 + \sigma^2 \Delta t, \tag{6.9}$$

with $\alpha = r - \frac{1}{2}\sigma^2$ as before, and the constraint on the kurtosis is dropped to allow h to be set separately. Solving the above yields

$$p_u(h, \Delta t) = \frac{1}{2}\left(\sigma^2 \frac{\Delta t}{h^2} + \alpha^2 \frac{\Delta t^2}{h^2} + \alpha \frac{\Delta t}{h}\right), \tag{6.10}$$

$$p_m(h, \Delta t) = 1 - p_u - p_d, \tag{6.11}$$

and

$$p_d(h, \Delta t) = \frac{1}{2}\left(\sigma^2 \frac{\Delta t}{h^2} + \alpha^2 \frac{\Delta t^2}{h^2} - \alpha \frac{\Delta t}{h}\right). \tag{6.12}$$

In the above specification, not all values of h will result in non-negative probabilities once Δt has been fixed. Since h and Δt are both usually much smaller than 1,

$$p_m = 1 - p_u - p_d \approx 1 - \sigma^2 \frac{\Delta t}{h^2}.$$

This yields the constraint

$$\sigma^2 \frac{\Delta t}{h^2} < 1,$$

or

$$\lambda > 1,$$

where $\lambda = \frac{h^2}{\sigma^2 \Delta t}$. In the case of the Gao1 parameterization, λ was equal to 3, which will serve as a target value for this model.

To aid a description of the implementation, it is worthwhile to refer to Figure 6.7. The primary difference between the original AMM and the adaption for barrier options is the flow of information. Instead of using a refined mesh to update the node values of the coarse mesh, the coarse mesh is positioned in such a way that the initial asset price coincides with the t_0-node of the refined mesh. Thus, the option is actually priced on the highest resolution mesh. The basic algorithm for a single layer of refinement is as follows[2]:

1. Construct a coarse Gao2 tree with the initial node a distance h above the barrier. This ensures that there is one coarse-mesh step between this initial node and the barrier. Determine the option value at each node in this tree.

[2] Please note that this a summary of the model when attaching a single higher resolution mesh. For the details of the full implementation, which is isomorphic at successive levels of refinement, please see (Figlewski and Gao, 1999).

2. Compute option values along the barrier and along the coarse mesh node layer above the barrier in time-steps of $\frac{\Delta t}{4}$. These are the nodes that will "anchor" the refined mesh to the tree. They are visible as red circles in Figure 6.7.

3. Using backward iteration, compute the option values along $\ln(B) + \frac{h}{2}$ at time-steps of $\frac{\Delta t}{4}$. These are the red stars visible in Figure 6.7 and they form the central node layer of the refined mesh. Their value is computed using the discounted expectation of the anchoring nodes. The resulting value at the t_0 refined mesh node will be the option value.

All that remains is to choose h and Δt in such a way that the tree is positioned correctly to price the option on the refined mesh and keep the probabilities positive. There is only a single value of h that will allow the initial node of the highest resolution mesh to coincide with the initial asset price,

$$h = 2^R(\ln S(0) - \ln H), \tag{6.13}$$

where R is number of refined lattices to attach to the coarse mesh. However, with λ fixed at 3, this can result in a Δt such that there are a non-integer number of total time-steps in the coarse tree, as $N = \frac{T}{\Delta t}$. To compensate, some flexibity must be allowed for in the value of λ. (Figlewski and Gao, 1999) propose simply rounding the resulting N-value and then using it to compute an updated Δt value. This results in slightly longer time-steps than would be the case if λ was fixed at 3, but it does ensure that every layer of mesh has an integer number of time-steps. The final size of the coarse-mesh time-step becomes

$$\Delta t = \frac{T}{\text{round}\left(\frac{\lambda\sigma^2}{h^2}T\right)}. \tag{6.14}$$

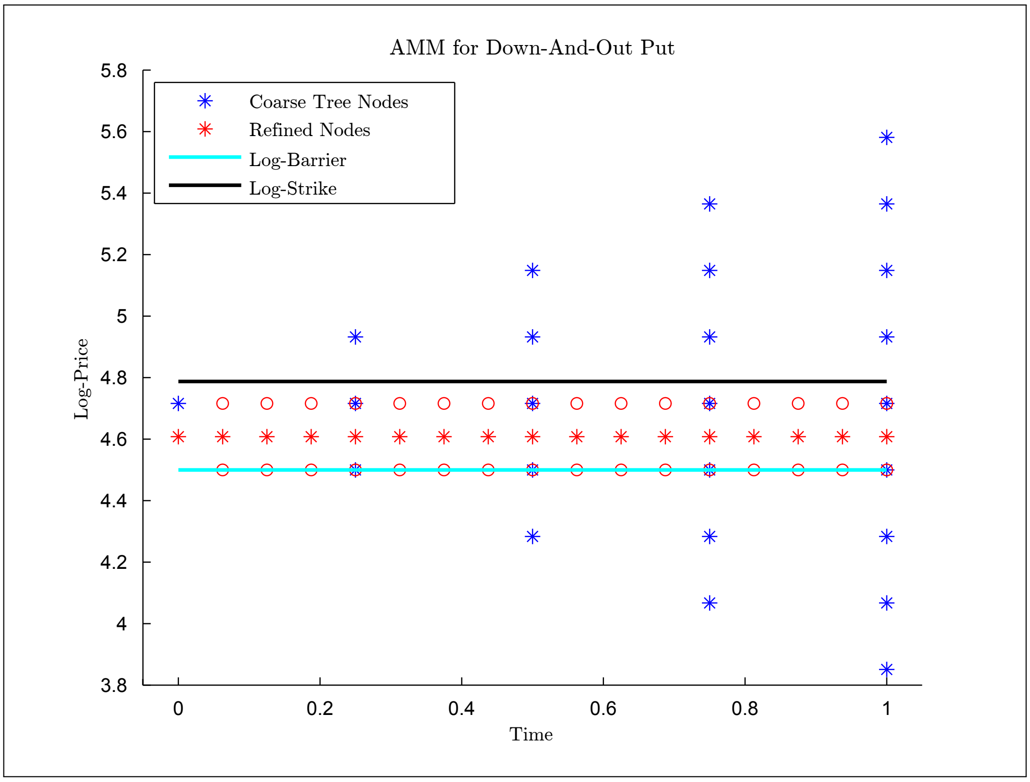
6.2.3 Results

Thirteen European Down-And-Out put options were priced with the parameter specification as given below:

Parameter:	**Distribution:**
$S_1(t_0)$	Uniformly spaced from 90.25 in steps of 0.25 until 93.25.
σ	Constant at 0.25.
r	Constant at 0.1.
T	Constant at 1.
K	Constant at 100.
B	Constant at 90.

These parameters are in line with those used to generate the test results in Table 3 of (Figlewski and Gao, 1999). These are extreme conditions for option pricing - for the smallest value of $S_1(t_0)$ the initial asset price is less than .3% above the knock-out barrier.

Fig. 6.7: The AMM for a Down-And-Out put with a single layer of refinement. The parameters are $S(0) = 95,\ K = 120,\ r = 0.1,\ \sigma = 0.25,\ T = 1$ and the barrier is set at 90.

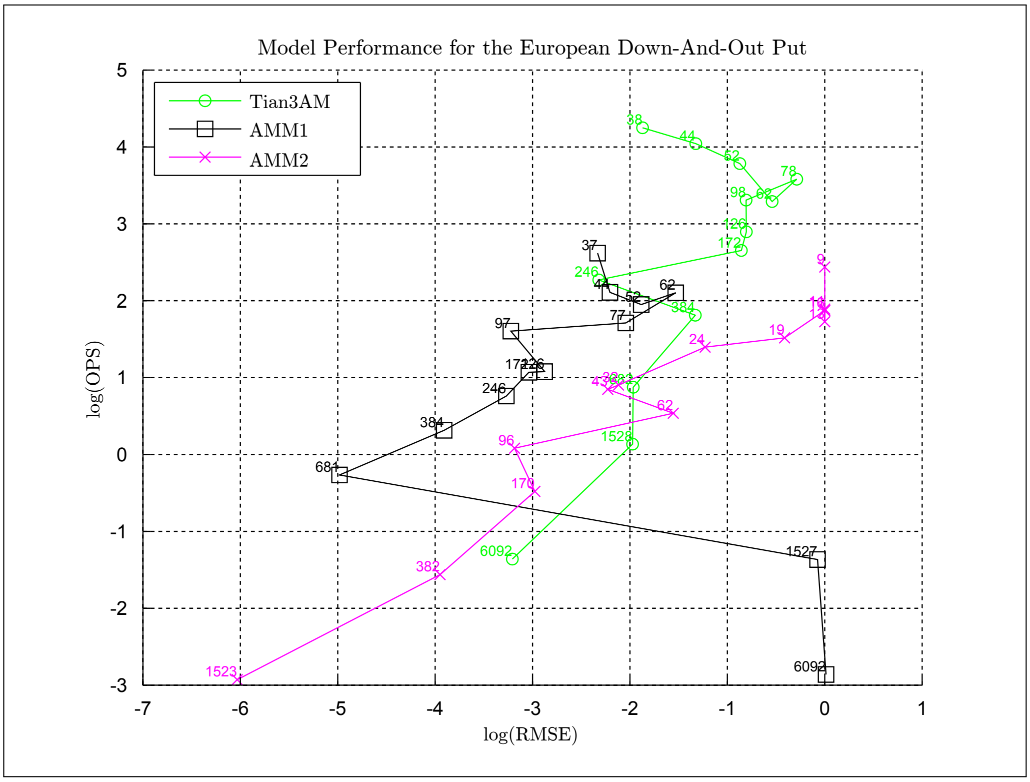


Since the number of time-steps in the Adaptive Mesh Model is dependent on the distance from the barrier and the level of refinement, the Tian3AM tree was generated based on the number of time-steps in the coarse mesh of the AMM1 model at each initial asset value. When necessary, N was increased by 1 to allow for an even number of steps in the Tian3AM so that Richardson extrapolation could be used. All options were priced fitfy times on each tree at each initial asset price and the average execution time was taken to determine the options priced per second. The root mean square relative error was computed as in Section 6.1 with the caveat that the analytical price of each option is known. The results are displayed in Figure 6.8 and Table 6.4.

By comparing Figures 6.8 and 6.1, it is immediately apparent how much more erratic the pricing of a barrier option is as compared to an American put. This comparison cannot be taken too far however, as Figure 6.1 shows the log-metrics when pricing the same set of options for ever increasing N-values, whereas each point in Figure 6.8 represents an option with a different initial asset value.

There were two points in the simulation where increasing the number of time-steps actually resulted in faster pricing, however the root mean square error increased as well. This is visible in Figure 6.8 at $N = 62$ for the AMM1 model and $N = 78$

Fig. 6.8: The log-metrics for thirteen European Down-And-Out put options priced on the three specified models. The number next to each data point indicates the number of time-steps used in that computation.



for the Tian3AM tree.

As was expected after examining the convergence of the standard Tian3 tree (see Figure 6.6), the error is very sensitive to the node-positioning. When the nodes are shifted slightly (by changing Δt for the Tian3AM tree or h for the AMM trees), the number of nodes in the tree at the terminal time that are in-the-money can increase or decrease dramatically and thus shift the lattice-generated option value.

For the highest four values of $S_1(t_0)$, the AMM2 tree fails to price the option reliably. This is somewhat unsurprising, as for those initial asset prices the resulting AMM2 tree has less than 20 time-steps. For the lowest two values of $S_1(t_0)$, the values closest to the barrier, the AMM1 tree performs very poorly. Despite the reasonable N-values in the tree, there is simply not enough pricing information captured in the space between the initial asset price and the knock-out barrier.

The AMM2 tree performs well in the region for which it was designed. Although slower than the Tian3AM tree with 6092 time-steps, it prices the option significantly more accurately. It is worth noting that at the lowest two initial asset prices, the true option value is less than 0.005.

Tab. 6.4: Comparison across the three tree methods when pricing uniformly spaced European Down-And-Out put options.

		Tian3AM		**AMM1**		**AMM2**	
$\mathbf{S_1(t_0)}$	**True Price**	RMS	OPS	RMS	OPS	RMS	OPS
93.25	2.9452e-02	1.5403e-01	70.0927	9.7106e-02	13.7202	9.9998e-01	11.4563
93	2.7343e-02	2.6613e-01	57.0092	1.1021e-01	8.2405	1.0000	6.6295
92.75	2.5206e-02	4.1518e-01	43.9917	1.5192e-01	7.0320	1.0000	5.6243
92.5	2.3042e-02	5.8257e-01	26.8521	2.1652e-01	8.1952	1.0000	6.5012
92.25	2.0851e-02	7.5021e-01	35.9016	1.2968e-01	5.5349	6.6211e-01	4.5590
92	1.8634e-02	4.4591e-01	27.3558	3.9832e-02	4.9766	2.9282e-01	4.0411
91.75	1.6390e-02	4.4700e-01	18.1059	5.6384e-02	2.9391	1.2035e-01	2.4686
91.5	1.4121e-02	4.2471e-01	14.2088	4.7948e-02	2.9136	1.0779e-01	2.3342
91.25	1.1827e-02	9.8264e-02	9.6857	3.8047e-02	2.1398	2.1107e-01	1.7104
91	9.5083e-03	2.6471e-01	6.1203	2.0088e-02	1.3699	4.1384e-02	1.0853
90.75	7.1657e-03	1.3997e-01	2.3982	6.8610e-03	0.7664	5.0974e-02	0.6186
90.5	4.7997e-03	1.3902e-01	1.1444	9.2863e-01	0.2551	1.9258e-02	0.2096
90.25	2.4109e-03	4.0458e-02	0.2566	1.0134	0.0572	2.4000e-03	0.0535

Chapter 7

Conclusion

Joshi (2007) and Chan *et al.* (2009) found that traditional tree methods implemented with the acceleration techniques of smoothing, Richardson extrapolation and truncation outperformed those without, when pricing an American put. This work has extended that analysis to show that the acceleration techniques aid the convergence of the Tian3 model even when pricing a Down-And-Out barrier option (as evidenced in Figure 6.6), as well as when implementing the Modified Barrier Algorithm.

The Adaptive Mesh Model outperformed the Tian3AM model for pricing the considered barrier option but failed to compete with the top performing, accelerated traditional tree method when pricing an American put. However, the simple measurements of speed and root mean squared error are not the only factors to consider. In truth, the AMM for barrier options is an entirely separate implementation and parameterization from the standard AMM and could be considered a separate model, despite the shared underlying paradigm of attaching a higher resolution mesh to a coarse tree. In addition, the AMM for barrier options is designed for the very specific case of when the initial asset price is quite close to the barrier. Should the distance in log-space between the initial asset price and the barrier become too large, the resulting AMM will contain too few nodes for accurate valuation.

In terms of implementation, both versions of the AMM require more development time than the traditional tree methods. The Tian3 and Tian4 models are both intuitive and simple to implement, although care must be taken when implementing the truncation technique as truncating the wrong nodes can result in an option valuation that is almost correct and thus this error can be difficult to detect.

In conclusion, the additional development time for the AMM is not rewarded when pricing the American put option, but the modification for barrier options performs significantly better than the Tian3AM model, while suffering from having far less applicability. As will always be the case, the selection of model and parameterization will ultimately depend on the purpose of the valuation and the circumstances of the implementation.

Bibliography

Andricopoulos, A. D., Widdicks, M., Duck, P. W. and Newton, D. P. (2004). Curtailing the range for lattice and grid methods, *The Journal of Derivatives* **11**(4): 55–61.

Björk, T. (2004). *Arbitrage theory in continuous time*, Oxford Finance Series, Oxford University Press, Incorporated.

Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities, *The journal of political economy* pp. 637–654.

Boyle, P. P. (1986). Option valuation using a three-jump process, *International Options Journal* **3**(7-12): 1.

Broadie, M. and Detemple, J. (1996). American option valuation: new bounds, approximations, and a comparison of existing methods, *Review of Financial Studies* **9**(4): 1211–1250.

Chan, J. H., Joshi, M., Tang, R. and Yang, C. (2009). Trinomial or binomial: Accelerating American put option price on trees, *Journal of Futures Markets* **29**(9): 826–839.

Chance, D. (2007). A synthesis of binomial option pricing models for lognormally distributed assets, *Available at SSRN 969834* .

Chen, T. and Joshi, M. (2012). Truncation and acceleration of the Tian tree for the pricing of American put options, *Quantitative Finance* **12**(11): 1695–1708.

Cox, J. C., Ross, S. A. and Rubinstein, M. (1979). Option pricing: A simplified approach, *Journal of financial Economics* **7**(3): 229–263.

Derman, E., Kani, I., Ergener, D. and Bardhan, I. (1995). Enhanced numerical methods for options with barriers, *Financial Analysts Journal* **51**(6): pp. 65–74.

Diener, F. and Diener, M. (2004). Asymptotics of the price oscillations of a European call option in a tree model, *Mathematical finance* **14**(2): 271–293.

Figlewski, S. and Gao, B. (1999). The Adaptive Mesh Model: A new approach to efficient option pricing, *Journal of Financial Economics* **53**(3): 313–351.

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*, Applications of mathematics: Stochastic modelling and applied probability, Springer.

Hsia, C. C. (1983). On binomial option pricing, *Journal of Financial Research* **6**(1): 41–46.

Hull, J. (2010). *Options, Futures, and Other Derivatives, 7/e (With CD)*, Pearson Education India.

Hull, J. and White, A. (1988). The use of the control variate technique in option pricing, *Journal of Financial and Quantitative Analysis* **23**(3): 237–251.

Jarrow, R. A. and Rudd, A. (1983). *Option pricing*, Irwin Series in Finance, Dow Jones-Irwin.

Jarrow, R. and Turnbull, S. (1996). *Derivative Securities*, Derivative Securities, South-Western College Publishing, International Thompson Publishing.

Joshi, M. (2007). The convergence of binomial trees for pricing the American put, *Available at SSRN 1030143* .

Leisen, D. P. (1998). Pricing the American put option: A detailed convergence analysis for binomial models, *Journal of Economic Dynamics and Control* **22**(8-9): 1419–1444.

Taylor, D. R. (2013). Introduction to Finance and Derivatives, M.Phil. Course, University of Cape Town, South Africa.

Tian, Y. (1993). A modified lattice approach to option pricing, *Journal of Futures Markets* **13**(5): 563–577.

Walsh, J. B. (2003). The rate of convergence of the binomial tree scheme, *Finance and Stochastics* **7**(3): 337–361.